The Modal Logic of Copy and Remove

Carlos Areces[a], Hans van Ditmarsch[b], Raul Fervari[a], François Schwarzentruber[c]

[a] *FaMAF, Universidad Nacional de Córdoba & CONICET, Argentina*
[b] *LORIA, CNRS - Université de Lorraine, France & IMSc, Chennai, India*
[c] *ENS Rennes, France*

Abstract

We propose a logic with the dynamic modal operators copy and remove. The copy operator replicates a given model, and the remove operator removes paths in a given model. We show that the product update by an action model in dynamic epistemic logic decomposes in copy and remove operations. We also show that copy and remove operators (of path of length 1) can be expressed by action models. We investigate the expressive power of the logic with copy and remove operations, together with the complexity of the satisfiability problem of some of its syntactic fragments.

Keywords: modal logic, dynamic epistemic logic, complexity, expressivity.

---

1. Introduction

In modal logic we interpret a modal operator by way of an accessibility relation in a given model. Over the past decades logics have been proposed in which the modality is, instead, interpreted by a transformation of the model. In such logics the modality can be seen as interpreted by a binary relation between pointed Kripke models, where the second argument of the relation is the transformed model. We could mention sabotage logic here [1], wherein states or arrows are deleted from a model. Or we could mention dynamic epistemic logics [2] that focus on such model changing operators in view of modeling change of knowledge or belief (the standard interpretation for the basic modalities in that setting). In [3, 4, 5] a new line of contributions to model-transforming logics, motivated by van Benthem's sabotage logic is developed. Our contribution advances that last line of work, while linking it to dynamic epistemic logics.

Action model logic ($\mathcal{AML}$) [6] is a well-known dynamic epistemic logic to model information change. $\mathcal{AML}$ is an extension of basic epistemic logic with

Email addresses: `areces@famaf.unc.edu.ar` (Carlos Areces), `hans.van-ditmarsch@loria.fr` (Hans van Ditmarsch), `fervari@famaf.unc.edu.ar` (Raul Fervari), `francois.schwarzentruber@ens-rennes.fr` (François Schwarzentruber)

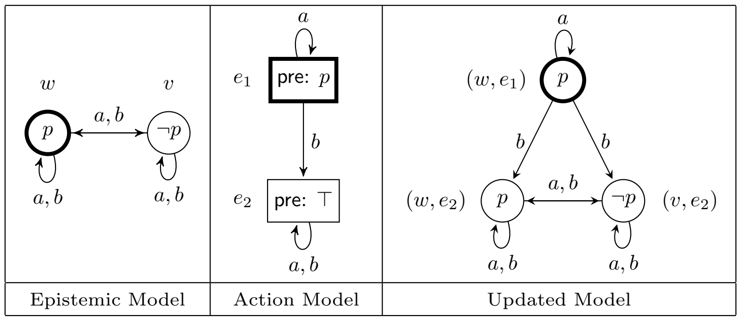


Figure 1: Agent a privately learns that p.

a dynamic modal operator for the execution of actions. This operator is parameterized by an action model, a semantic object which typically models a multi-agent information changing scenario. These actions models are treated as syntactic objects in modal operators. Action models are complex structures, which also leads to high computational complexity (deciding model checking is PSPACE-complete, while deciding satisfiability is NEXPTIME-complete [7]).

In this contribution we propose modal logics with primitive actions called copy and remove. We investigate some of their model theoretic properties and their complexity, and, as an example of what one can do with such logics, we give an embedding of action model logic into our logic: we show that every action model (with propositional pre-conditions) can be simulated by a combination of the copy and remove operators. This is in line with the previously known result that, on the class of finite models, action model execution corresponds to model restriction ('remove') on a bisimilar copy ('copy') of the initial model [8]. The delete we propose is akin to the generalized arrow updates of [9], continuing the work started in [10], that are also known to have equal expressivity as action model logic. But the copy and remove operators we propose are more procedural, whereas these mentioned results are more of a declarative nature.

In Figure 1 we show an epistemic model (a Kripke model), an action model, and the result of executing that action model in that epistemic model. The epistemic model represents that agents a and b are uncertain whether an atomic proposition p is true (and that they have common knowledge of that uncertainty). The actual world, or designated state, of the model is where p is true (shown with a thick circle in the figure). The action model represents that agent a learns that p is true, whereas agent b (incorrectly) believes that nothing happens—of which a is aware. In short: a privately learns that p. In action models, the valuations of propositional variables are replaced by pre-conditions, in this case p and $\top$ (the formula that is always true). Action models update Kripke models by mean of a restricted modal product, where the domain is limited to the state-action pairs where the pre-conditions of the actions hold. Therefore, there are only three (and not four) pairs in the updated model: the pair (v, e_1) is missing as the pre-condition of e_1, the formula p, is not true in

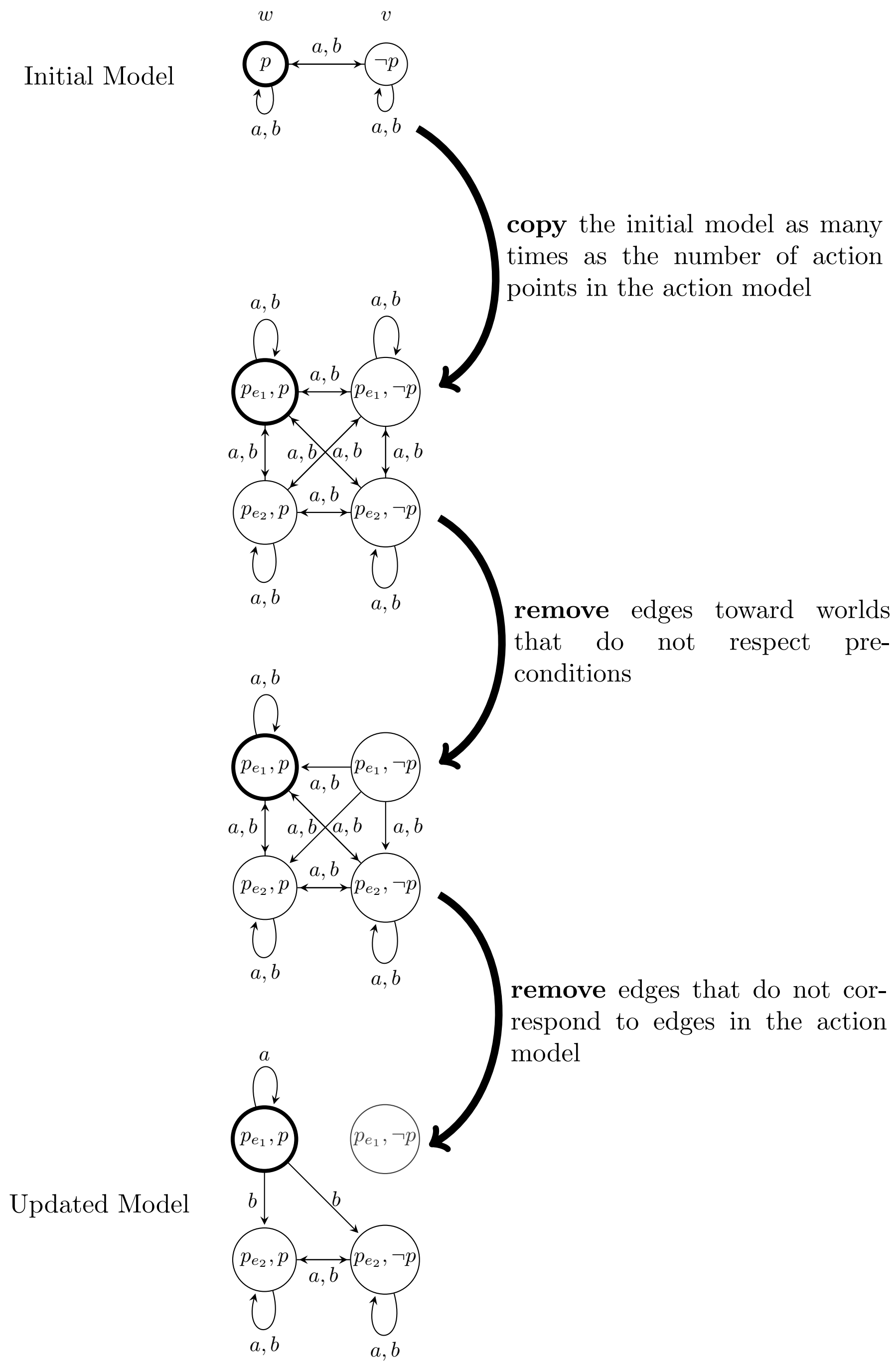


Figure 2: The result of one copy and two remove operations on the epistemic model of Figure 1, again resulting in the same (bisimilar) updated model.

state v. The arrows in the restricted product are updated according to the principle that there is a (labeled) arrow between two state-action pairs if there was such an arrow linking both the first arguments and the second arguments. One can now establish that in the resulting model a knows that p (there is only an a-arrow from w to itself), whereas b still believes that a, b are ignorant about p.

By means of the copy and remove actions of the logics that we propose, we can alternatively describe the effect of this action model. This is depicted in Figure 2. First, we replicate the original epistemic model as many times as there are actions in the action model (twice in this case). We identify each copy with a (fresh) propositional variable corresponding to an action in the action model (e.g., p_{e_1} corresponds to e_1). Thus we obtain the second model in Figure 2. Then, we first remove all the edges (arrows) that point to state-action alternatives wherein the action cannot be executed in the state. Finally, between the remaining state-action pairs we remove all edges that are ruled out according to the accessibility relation in the action model. Thus we obtain the "updated model" in Figure 2. Together with some of the results presented in this paper, the logics with copy and remove were first introduced in [11]. In this article, we extend these results, and give more detailed proofs.

In Section 2 we introduce the formal definition of action model logic, together with an example which motivates the use of the fragment with only Boolean pre-conditions. In Section 3 we introduce $\mathcal{ML}(\mathsf{cp}, \mathsf{rm})$, the logic with the two dynamic primitives copy and remove which captures the behaviour of action models. In Section 4 we introduce bisimulations to investigate its expressive power. In Section 5 we define an equivalence preserving translation from action model logic to a fragment of $\mathcal{ML}(\mathsf{cp}, \mathsf{rm})$. We also show that it is possible to find action models that encode copy and remove actions, which give us a method to decompose action models with Boolean pre-conditions. Finally, we show complexity results for different fragments of $\mathcal{ML}(\mathsf{cp}, \mathsf{rm})$ in Section 6.

2. Action Model Logic with Boolean Pre-conditions

One of the main results in this paper consists in showing that we can capture action model logic with simpler primitives, as we mentioned in the introduction. Let us start by introducing formally the logic $\mathcal{AML}$. First, we introduce action models.

In what follows let PROP be an infinite countable set of propositional symbols, and AGT a finite set of agents symbols disjoint from PROP.

Definition 2.1 (Action Models). *Let $\mathcal{B}$ be the classical propositional language over PROP. An action model $\mathcal{E}$ is a structure $\mathcal{E} = \langle E, \rightarrow, \mathsf{pre}, \mathsf{post}\rangle$, where E is a non-empty finite set whose elements are called action points. For each $a \in \mathsf{AGT}$, $\rightarrow_a \subseteq E \times E$ is an equivalence relation; $\mathsf{pre} : E \rightarrow \mathcal{B}$ is a pre-condition function; and $\mathsf{post} : E \rightarrow \mathsf{PROP} \rightarrow \mathcal{B}$ is a post-condition function. Let e be an action point in $\mathcal{E}$, the pair $(\mathcal{E}, e)$ is a pointed action model.*

Action models in action model logic appear as modalities. We introduce the syntax of $\mathcal{AML}$.

Definition 2.2 (Syntax). *The set* FORM *of formulas of* $\mathcal{AML}^+$ *over* PROP *and* AGT *is defined as:*

$$\mathsf{FORM} ::= \top \mid p \mid \neg\varphi \mid \varphi \wedge \psi \mid \Box_a\varphi \mid [\mathcal{E}, e]\varphi,$$

where $p \in \mathsf{PROP}$*,* $a \in \mathsf{AGT}$*,* $\varphi, \psi \in \mathsf{FORM}$*, and* $\mathcal{E}, e$ *is a pointed action model.*

We will call $\mathcal{ML}$ the fragment without the dynamic operator $[\alpha]$, and $\mathcal{AML}$ the fragment where post-conditions functions $\mathsf{post} : E \to \mathsf{PROP} \to \mathcal{B}$ of action models appearing in formulas are of the form $\mathsf{post}(e, p) = p$ for all action points $e \in E$. Intuitively, it corresponds to the fragment where action models have no post-condition and we may remove the post-conditions from the definition, i.e., we consider only action models of the shape $\langle E, \to, \mathsf{pre}\rangle$. $\langle\alpha\rangle\varphi$ is a shorthand for $\neg[\alpha]\neg\varphi$, $\Diamond_a\varphi$ for $\neg\Box_a\neg\varphi$, and other Boolean operators are defined as usual.

Formulas of action model logic are interpreted in (pointed) models.

Definition 2.3 (Models). *A model* $\mathcal{M}$ *is a triple* $\mathcal{M} = \langle W, R, V\rangle$*, where* W *is a non-empty set;* $R \subseteq \mathsf{AGT} \times W^2$ *is an accessibility relation (we will often write* R_a *to refer to the set* $\{(w, v) \in W^2 \mid (a, w, v) \in R\}$*); and* $V : \mathsf{PROP} \to 2^W$ *is a valuation. A pair* $\mathcal{M}, w$ *where* w *is a state in* $\mathcal{M}$ *is called a pointed model.*

Now we can introduce the semantics of action model logic.

Definition 2.4 (Semantics). *Given a pointed model* $\mathcal{M}, w$ *with* $\mathcal{M}=\langle W, R, V\rangle$*, an action pointed model* $\mathcal{E}, e$ *with* $\mathcal{E} = \langle E, \to, \mathsf{pre}, \mathsf{post}\rangle$*, and a formula* φ *we say that* $\mathcal{M}, w \models \varphi$ *when*

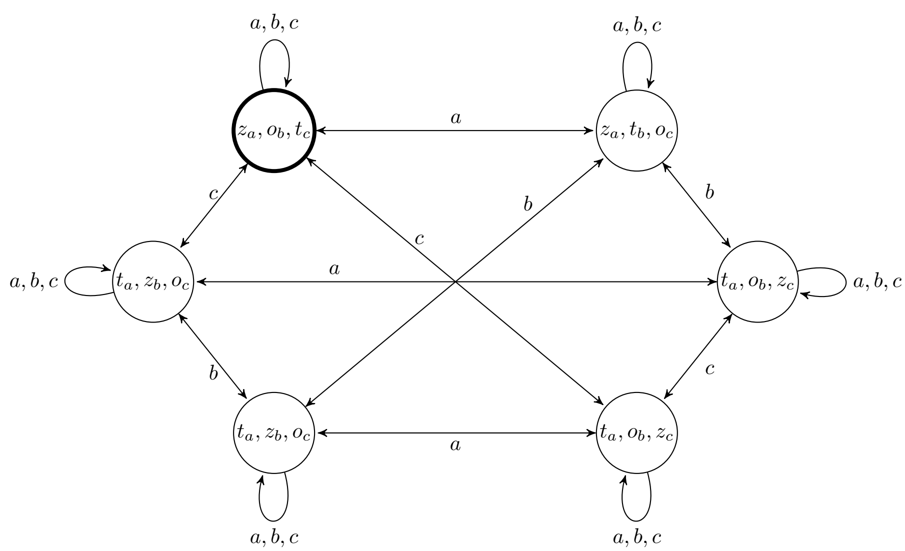
$$\begin{array}{lll}
\mathcal{M}, w \models \top & & \textit{always} \\
\mathcal{M}, w \models p & \textit{iff} & w \in V(p) \\
\mathcal{M}, w \models \neg\varphi & \textit{iff} & \mathcal{M}, w \not\models \varphi \\
\mathcal{M}, w \models \varphi \wedge \psi & \textit{iff} & \mathcal{M}, w \models \varphi \textit{ and } \mathcal{M}, w \models \psi \\
\mathcal{M}, w \models \Box_a\varphi & \textit{iff} & \textit{for all } v \in W \textit{ s.t. } (w, v) \in R_a,\ \mathcal{M}, v \models \varphi \\
\mathcal{M}, w \models [\mathcal{E}, e]\varphi & \textit{iff} & \mathcal{M}, w \models \mathsf{pre}(e) \textit{ implies } (\mathcal{M} \otimes \mathcal{E}), (w, e) \models \varphi,
\end{array}$$

where the restricted product $(\mathcal{M} \otimes \mathcal{E})$ *is defined as* $\langle W^{\otimes}, R^{\otimes}, V^{\otimes}\rangle$*, with:*

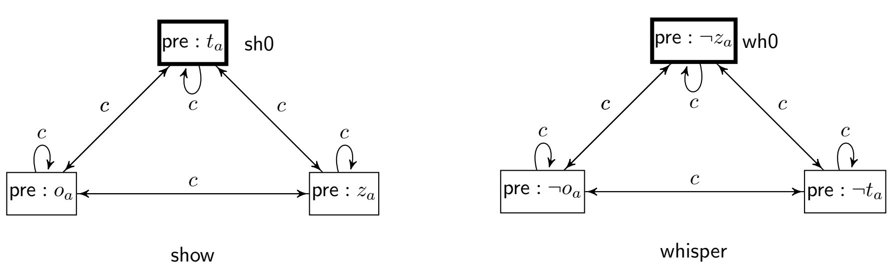
$$\begin{array}{lll}
W^{\otimes} & = & \{(v, d) \in W \times E \mid \mathcal{M}, v \models \mathsf{pre}(d)\} \\
((v, d), (u, f)) \in R^{\otimes}_a & \textit{iff} & (v, u) \in R_a \textit{ and } d \to_a f \\
V^{\otimes}(p) & = & \{(v, d) \mid \mathcal{M}, v \models \mathsf{post}(d)(p)\}.
\end{array}$$

It is known that the notion of bisimulation needed for $\mathcal{AML}$ coincides with the one for $\mathcal{ML}$. Notice that we have not introduced the full action model logic as in [6]: we only consider Boolean pre and post-conditions in action models. The logic with copy and remove that we will introduce in the next section, only captures this fragment. However, even under this restriction, the expressive power of the logic is sufficient to capture interesting epistemic scenarios.

Example 2.5. *Consider the following classical example in epistemic logics literature. There are three agents: Anne (a), Bill (b) and Cath (c); each of them holds one of three possible cards: zero, one or two. Propositional symbols such as z_i, o_i and t_i state that the agent i is holding the card (z)ero, (o)ne and (t)wo, respectively. The following model shows all the possible situations. Each agent is uncertain about any other agent's card, which is also modeled in the figure. For instance, Anne cannot distinguish the two states at the top of the figure (in both states z_a holds, i.e., Anne has card zero but she does not know if Bob has card one and Cath has card two or the other way round). But she can obviously distinguish these states of any other (in which she has a different card).*



Let us consider the scenario in which Anne shows *card zero to Bill. This is represented by the leftmost action model below. Cath cannot see the face of the shown card, but notices that a card is being shown. The action model shown on the right codifies the situation where Anne tells Bill a card that she does not have. Anne* whispers *in Bill's ear "I do not have card two". Cath notices that Anne reveals she does not have some card, but cannot hear which card.*



3. Copy and Remove

In this section we introduce $\mathcal{ML}(\mathsf{cp}, \mathsf{rm})$, a language which can create copies of a model and remove edges. We start by introducing its formal syntax.

Definition 3.1 (Syntax). *Define the set* FORM *of* $\mathcal{ML}(\mathsf{cp}, \mathsf{rm})$*-formulas, together with a set* PATH *of path expressions.*

$$\mathsf{FORM} ::= \top \mid p \mid \neg\varphi \mid \varphi \wedge \varphi' \mid \Box_a \varphi \mid \mathsf{rm}(\pi)\varphi \mid \mathsf{cp}(Q, q)\varphi,$$

where Q is any finite set of propositional symbols, $q \in Q$, $a \in \mathsf{AGT}, \varphi, \varphi' \in \mathsf{FORM}$, and $\pi \in \mathsf{PATH}$.

$$\mathsf{PATH} ::= a \mid \pi; \pi' \mid \varphi?,$$

where $a \in \mathsf{AGT}$, $\pi, \pi' \in \mathsf{PATH}$ and φ is a Boolean formula.

The operation $\mathsf{rm}(\pi)\varphi$ is related to the sabotage operator introduced by van Benthem in [1] (investigated also in [12, 13, 3, 5]), and its intuitive meaning is that φ holds after having deleted all edges that appear in paths that match π. The intuitive meaning of the operator $\mathsf{cp}(Q, q)\varphi$ is that after replicating the initial model, φ is true in the component labeled by q.

We also define the following syntactic fragments: $\mathcal{ML}(\mathsf{cp})$, the fragment with only the cp operator; $\mathcal{ML}(\mathsf{rm})$, the fragment with only rm; $\mathcal{ML}(\mathsf{rm}^1)$, the fragment with only rm restricted to path expressions of the form $\pi = \varphi?; a; \psi?$; and $\mathcal{ML}(\mathsf{cp}, \mathsf{rm}^1)$, the fragment with rm with path expressions only of the form $\pi = \varphi?; a; \psi?$ and with cp.

Formulas of $\mathcal{ML}(\mathsf{cp}, \mathsf{rm})$ are interpreted over models such as those introduced in Definition 2.3. We now define the satisfaction relation.

Definition 3.2 (Paths). *We represent a path as a sequence $w_0 a_0 w_1 a_1 \ldots w_{n-1} a_{n-1} w_n$ where w_i are states and a_i are agents. Let $\mathcal{M} = \langle W, R, V\rangle$ be a model, $\pi, \pi' \in \mathsf{PATH}$. We define the set of π-paths $\mathcal{P}^{\mathcal{M}}(\pi)$ of $\mathcal{M}$ by induction on π as follows:*

$$\begin{array}{lcl}
\mathcal{P}^{\mathcal{M}}(a) & = & \{wau \mid (w, u) \in R_a\} \\
\mathcal{P}^{\mathcal{M}}(\pi; \pi') & = & \{SwS' \mid Sw \in \mathcal{P}^{\mathcal{M}}(\pi) \text{ and } wS' \in \mathcal{P}^{\mathcal{M}}(\pi')\} \\
\mathcal{P}^{\mathcal{M}}(\varphi?) & = & \{w \mid \mathcal{M}, w \models \varphi\}.
\end{array}$$

Let $a \in \mathsf{AGT}$ and P a path, we define $edges_a(P)$ as the set of a-edges in P. Formally, $edges_a(P) = \{(a, w, u) \mid wau \text{ is a subsequence of } P\}$.

Definition 3.3 (Updated Models). *Given a model $\mathcal{M} = \langle W, R, V\rangle$, a path expression π, and $Q \subseteq \mathsf{PROP}$, we define the updated models*

$\mathcal{M}_{\mathsf{rm}(\pi)} = \langle W, R_{\mathsf{rm}(\pi)}, V\rangle$, *where*

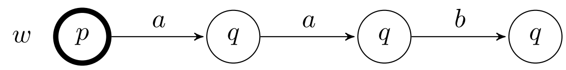
$$R_{\mathsf{rm}(\pi)} = R \setminus \bigcup_{a \in \mathsf{AGT}, P \in \mathcal{P}^{\mathcal{M}}(\pi)} edges_a(P).$$

$\mathcal{M}_{\mathsf{cp}(Q)} = \langle W_{\mathsf{cp}(Q)}, R_{\mathsf{cp}(Q)}, V_{\mathsf{cp}(Q)}\rangle$, *where*

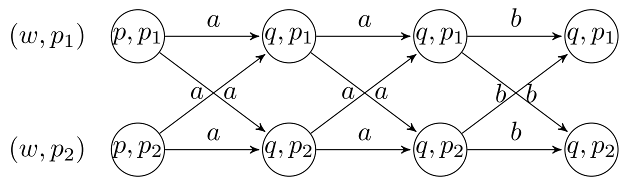
$$\begin{array}{lcl}
W_{\mathsf{cp}(Q)} & = & \{(w, q) \mid w \in W \text{ and } q \in Q\} \\
R_{\mathsf{cp}(Q)} & = & \{(a, (w, q), (w', q')) \mid (a, w, w') \in R\} \\
V_{\mathsf{cp}(Q)}(p) & = & \{(w, q) \mid w \in V(p)\} \text{ for } p \neq q \\
V_{\mathsf{cp}(Q)}(q) & = & \{(w, q) \mid w \in W\}.
\end{array}$$

We will discuss some examples of updated models.

Example 3.4. *Let us consider the following model $\mathcal{M}$:*

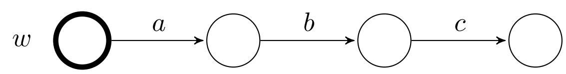


The updated model $\mathcal{M}_{\mathsf{cp}(\{p_1,p_2\})}$ is shown below. We replicate the original model in two copies: one satisfying p_1 and another satisfying p_2.

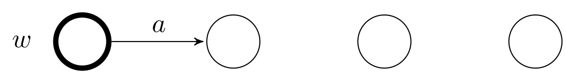


Notice that $\mathcal{M}_{\mathsf{cp}(\{p_1,p_2\})}$ does not contain new information about successors: each copy of a successor in the original model is also a successor in the new model.

Example 3.5. *Let us consider the following model $\mathcal{M}$:*

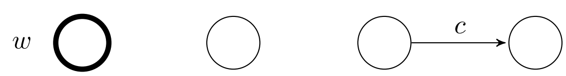


The updated model $(\mathcal{M}_{\mathsf{rm}(b;c)})_{\mathsf{rm}(a;b)}$ is:



The removal of $a;b$-paths from $\mathcal{M}_{\mathsf{rm}(b;c)}$ results in $\mathcal{M}_{\mathsf{rm}(b;c)}$ itself, because there are no $a;b$-paths in the model.

On the other hand, the model $(\mathcal{M}_{\mathsf{rm}(a;b)})_{\mathsf{rm}(b;c)}$ is:



We are ready to define the semantics of $\mathcal{ML}(\mathsf{cp},\mathsf{rm})$.

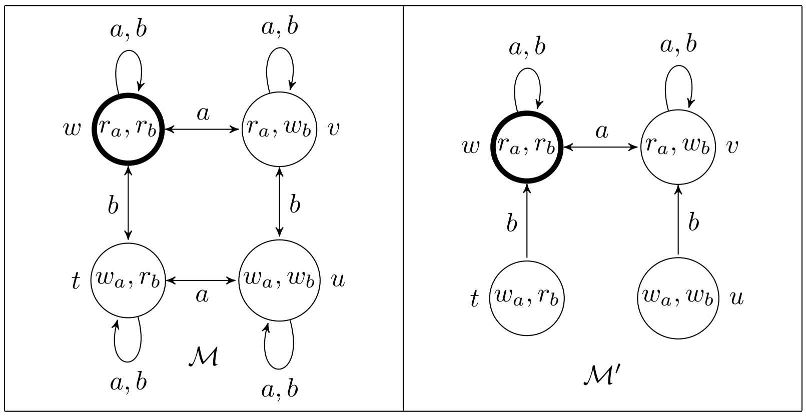
Definition 3.6 (Semantics). *Given a pointed model $\mathcal{M}, w$ and a formula φ we say that $\mathcal{M}, w$ satisfies φ, and write $\mathcal{M}, w \models \varphi$, when*

$$\begin{array}{lll}
\mathcal{M}, w \models p & \textit{iff} & w \in V(p) \\
\mathcal{M}, w \models \neg\varphi & \textit{iff} & \mathcal{M}, w \not\models \varphi \\
\mathcal{M}, w \models \varphi \wedge \psi & \textit{iff} & \mathcal{M}, w \models \varphi \textit{ and } \mathcal{M}, w \models \psi \\
\mathcal{M}, w \models \Box_a\varphi & \textit{iff} & \textit{for all } v \in W \textit{ s.t. } (w, v) \in R_a,\ \mathcal{M}, v \models \varphi \\
\mathcal{M}, w \models \mathsf{rm}(\pi)\varphi & \textit{iff} & \mathcal{M}_{\mathsf{rm}(\pi)}, w \models \varphi \\
\mathcal{M}, w \models \mathsf{cp}(Q, q)\varphi & \textit{iff} & \mathcal{M}_{\mathsf{cp}(Q)}, (w, q) \models \varphi.
\end{array}$$

The formula φ is satisfiable if for some pointed model $\mathcal{M}, w$ we have $\mathcal{M}, w \models \varphi$. We further define $\mathsf{cp}(Q)\varphi$ as an abbreviation for $\bigwedge_{q \in Q} \mathsf{cp}(Q, q)\varphi$. We usually write $\mathsf{cp}(p_1, \ldots, p_n)$ instead of $\mathsf{cp}(\{p_1, \ldots, p_n\})$.

Let us see an example of how rm can be useful to make announcements.

Example 3.7. *Suppose we are modeling a card game scenario, and we have the model $\mathcal{M}$ shown on the left, where agents a and b are holding either a red (r) or a white (w) card and where both are uncertain about the other agent's card.*

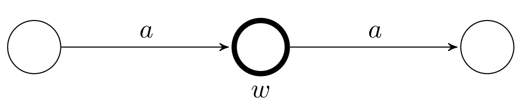


Now consider that agent a reveals to agent b that she has a red card. Modeling this with a public announcements [14], we remove the access to all the states where agent a does not have a red card. The formula $\mathsf{rm}(b;\neg r_a?)\top$ captures exactly this epistemic update, obtaining model $\mathcal{M}'$ shown on the right. After executing $\mathsf{rm}(b;\neg r_a?)\mathsf{rm}(a;\neg r_a?)$, agent b knows that agent a has a red card, i.e., $\mathcal{M}', w \models \mathsf{rm}(b;\neg r_a?)\mathsf{rm}(a;\neg r_a?)\Box_b r_a$.

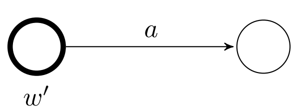
4. Expressive Power

Bisimulation is a classical notion introduced to investigate the expressive power of modal languages. The conditions required for a notion of bisimulation that captures the expressive power of $\mathcal{ML}(\mathsf{cp},\mathsf{rm})$ are very natural: paths deleted via rm traversing a particular state are characterized by the information in successors and predecessors of such point. Let us see an example.

Example 4.1. *Consider the formula $\mathsf{rm}(a;a)\Diamond_a\top$ and the following model $\mathcal{M}$:*



The formula does not hold at $\mathcal{M}, w$ because we delete all the paths matching two consecutive occurrences of an a-edge. On the other hand, the formula holds in the following model:



As we can see, path expressions can describe edges that arrive to the evaluation point. It turns out that $\mathcal{ML}(\mathsf{cp},\mathsf{rm})$-bisimulations are defined by the same conditions that defines bisimulations for the basic modal logic extended with the converse (past) modality $\Box^{-1}$, $\mathcal{ML}(\Box^{-1})$ (see [15]). $\mathcal{ML}(\Box^{-1})$ is the basic modal logic $\mathcal{ML}$ extended with the past operator $\Box^{-1}$. Let $\mathcal{M} = \langle W, R, V\rangle$ be a model, the semantics of $\Box^{-1}$ is defined as:

$$\mathcal{M}, w \models \Box_a^{-1}\varphi \text{ iff } \text{ for all } v \in W \text{ s.t. } (v,w) \in R_a,\ \mathcal{M}, v \models \varphi.$$

The formula $\Diamond_a^{-1}\varphi$ is a shorthand for $\neg\Box^{-1}\neg\varphi$.

Definition 4.2 (Bisimulations). *Let $\mathcal{M} = \langle W, R, V\rangle$ and $\mathcal{M}' = \langle W', R', V'\rangle$ be two models. A non empty relation $Z \subseteq W \times W'$ is an $\mathcal{ML}(\mathsf{cp},\mathsf{rm})$-bisimulation if it satisfies the following conditions. If wZw' then*

(atomic harmony) *for all $p \in \mathsf{PROP}$, $w \in V(p)$ iff $w' \in V'(p)$;*

(zig) *if $(w, v) \in R_a$ then for some v', $(w', v') \in R'_a$ and vZv';*

(zag) *if $(w', v') \in R'_a$ then for some v, $(w, v) \in R_a$ and vZv'.*

(zig^{-1}) *if $(v, w) \in R_a$ then for some v', $(v', w') \in R'_a$ and vZv';*

(zag^{-1}) *if $(v', w') \in R'_a$ then for some v, $(v, w) \in R_a$ and vZv'.*

We say that two pointed models $\mathcal{M}, w$ and $\mathcal{M}', w'$ are $\mathcal{ML}(\mathsf{cp},\mathsf{rm})$ (and write $\mathcal{M}, w \leftrightarrows_{\mathcal{ML}(\mathsf{cp},\mathsf{rm})} \mathcal{M}', w'$) if there is a $\mathcal{ML}(\mathsf{cp},\mathsf{rm})$-bisimulation Z such that wZw'.

Let $P \subseteq \mathsf{PROP}$, we say that two pointed models $\mathcal{M}, w$ and $\mathcal{M}', w'$ are P-bisimilar (and write $\mathcal{M}, w \leftrightarrows^P_{\mathcal{ML}(\mathsf{cp},\mathsf{rm})} \mathcal{M}', w'$) if there is a relation Z with wZw' satisfying the conditions for $\mathcal{ML}(\mathsf{cp},\mathsf{rm})$-bisimulations except that (atomic harmony) is restricted to propositional symbols occurring in P.

We can prove that $\mathcal{ML}(\mathsf{cp},\mathsf{rm})$-bisimilar models satisfy the same formulas. Without loss of generality we assume that all remove operators have the form $\mathsf{rm}(\varphi_1?; a_1; \varphi_2?; a_2; \ldots; a_{n-1}; \varphi_n?)\psi$, where $\varphi_i?$ are Boolean formulas, and $a_i \in \mathsf{AGT}$.

Proposition 4.3. *For all $\varphi \in \mathcal{ML}(\mathsf{cp},\mathsf{rm})$, there exists $\varphi' \in \mathcal{ML}(\mathsf{cp},\mathsf{rm})$ equivalent to φ, where the path expressions have the form $\varphi_1?; a_1; \varphi_2?; a_2; \ldots; a_{n-1}; \varphi_n?$, where $\varphi_i?$ are Boolean formulas, and $a_i \in \mathsf{AGT}$.*

Proof. The proposition is a consequence of the following equivalences which can be easily checked.

1. $\mathsf{rm}(\pi; \varphi_1?; \varphi_2?; \pi') \leftrightarrow \mathsf{rm}(\pi; \varphi_1? \wedge \varphi_2?; \pi')$.
2. $\mathsf{rm}(\pi; a_1; a_2; \pi') \leftrightarrow \mathsf{rm}(\pi; a_1; \top?; a_2; \pi')$.

These equivalences can be easily checked. $\Box$

We introduce three lemmas that will be helpful in the proof of Theorem 4.7.

Lemma 4.4. *Let $\mathcal{M}=\langle W, R, V\rangle$ and $\mathcal{M}'=\langle W', R', V'\rangle$ be models, $w \in W$, $w' \in W'$, be such that $\mathcal{M}, w \leftrightarrows_{\mathcal{ML}(\mathsf{cp},\mathsf{rm})} \mathcal{M}', w'$, and $\pi=\varphi_1?; a_1; \varphi_2?; \ldots; a_{n-1}; \varphi_n?$. Then, for all $P \in \mathcal{P}^{\mathcal{M}}(\pi)$ such that $P = w_0 a_0 \ldots w a_i \ldots w_n$, there is some $P' \in \mathcal{P}^{\mathcal{M}'}(\pi)$, with $P' = w'_0 a_0 \ldots w' a_i \ldots w'_n$ and for all $j \in \{1, \ldots, n\}$ we have $\mathcal{M}, w_j \leftrightarrows_{\mathcal{ML}(\mathsf{cp},\mathsf{rm})} \mathcal{M}', w'_j$.*

Proof. Given some $P \in \mathcal{P}^{\mathcal{M}}(\pi)$, we need to find $P' \in \mathcal{P}^{\mathcal{M}'}(\pi)$ satisfying the lemma. Suppose $P = w_0 a_0 \ldots w a_i \ldots w_n$. As $w a_i w_{i+1}$ is a subpath of P, $(w, w_{i+1}) \in R_{a_i}$. Because $\mathcal{M}, w \leftrightarrows_{\mathcal{ML}(\mathsf{cp},\mathsf{rm})} \mathcal{M}', w'$, by (zig) there is w'_{i+1} such that $(w', w'_{i+1}) \in R'_{a_i}$ and $\mathcal{M}, w_{i+1} \leftrightarrows_{\mathcal{ML}(\mathsf{cp},\mathsf{rm})} \mathcal{M}', w'_{i+1}$. For this reason, $\mathcal{M}, w_{i+1} \models \psi$ if and only if $\mathcal{M}', w'_{i+1} \models \psi$, for all ψ Boolean. Then, w_{i+1} is a good choice in order to construct P'. We can repeat this process to build the subpath $w' a_i w'_{i+1} \ldots w'_n$. In order to choose w_{i-1}, we can proceed in the same way but using (zig^{-1}), and repeating the process until we reach w'_1. Putting all together, we have constructed the right P'. □

Lemma 4.5. *Let $\mathcal{M} = \langle W, R, V\rangle$ and $\mathcal{M}' = \langle W', R', V'\rangle$ be models, $w \in W$, $w' \in W'$ and π a path expression. Then $\mathcal{M}, w \leftrightarrows_{\mathcal{ML}(\mathsf{cp},\mathsf{rm})} \mathcal{M}', w'$ implies $\mathcal{M}_{\mathsf{rm}(\pi)}, w \leftrightarrows_{\mathcal{ML}(\mathsf{cp},\mathsf{rm})} \mathcal{M}'_{\mathsf{rm}(\pi)}, w'$.*

Proof. We have to define a bisimulation $Z \subseteq W_{\mathsf{rm}(\pi)} \times W'_{\mathsf{rm}(\pi)}$. Define $Z = \{(v, v') \mid \mathcal{M}, v \leftrightarrows_{\mathcal{ML}(\mathsf{cp},\mathsf{rm})} \mathcal{M}', v'\}$. As $\mathcal{M}, w \leftrightarrows_{\mathcal{ML}(\mathsf{cp},\mathsf{rm})} \mathcal{M}', w'$, by Lemma 4.4 we know that each time we remove a path in $\mathcal{M}$, we also remove a path in $\mathcal{M}'$. Then immediately follows that Z is a bisimulation. □

Lemma 4.6. *Let $\mathcal{M} = \langle W, R, V\rangle$ and $\mathcal{M}' = \langle W', R', V'\rangle$ be models, $w \in W$ and $w' \in W'$. Then $\mathcal{M}, w \leftrightarrows_{\mathcal{ML}(\mathsf{cp},\mathsf{rm})} \mathcal{M}', w'$ implies $\mathcal{M}_{\mathsf{cp}(Q)}, (w, q) \leftrightarrows_{\mathcal{ML}(\mathsf{cp},\mathsf{rm})} \mathcal{M}'_{\mathsf{cp}(Q)}, (w', q)$.*

Proof. We have to define a bisimulation $Z \subseteq W_{\mathsf{cp}(Q)} \times W'_{\mathsf{cp}(Q)}$. Define:

$$Z = \{((v, q), (v', q)) \mid (v, q), (v', q) \in W_{\mathsf{cp}(Q)}, \text{ s.t. } \mathcal{M}, v \leftrightarrows_{\mathcal{ML}(\mathsf{cp},\mathsf{rm})} \mathcal{M}', v'\}.$$

(atomic harmony) holds because $(v, q) Z (v', q)$ iff v and v' satisfy (atomic harmony) in the original models, and (v, q) and (v', q) are both labeled by the symbol q. The required (zig) and (zag) conditions follow from the bisimulation between $\mathcal{M}$ and $\mathcal{M}'$ and the definition of $\mathcal{M}_{\mathsf{cp}(Q)}$ and $\mathcal{M}'_{\mathsf{cp}(Q)}$. □

Then we can state:

Theorem 4.7 (Invariance under bisimulation.). *For all $\varphi \in \mathcal{ML}(\mathsf{cp}, \mathsf{rm})$, $\mathcal{M}, w \leftrightarrows_{\mathcal{ML}(\mathsf{cp},\mathsf{rm})} \mathcal{M}', w'$ implies ($\mathcal{M}, w \models \varphi$ iff $\mathcal{M}', w' \models \varphi$).*

Proof. The proof is by structural induction. Let $\mathcal{M} = \langle W, R, V\rangle$ and $\mathcal{M}' = \langle W', R', V'\rangle$ be such that $\mathcal{M}, w \leftrightarrows_{\mathcal{ML}(\mathsf{cp},\mathsf{rm})} \mathcal{M}', w'$. We only prove the inductive cases for rm and cp.

rm(π)φ: Suppose $\mathcal{M}, w \models \mathsf{rm}(\pi)\varphi$, then $\mathcal{M}_{\mathsf{rm}(\pi)}, w \models \varphi$. By hypothesis $\mathcal{M}, w \leftrightarroweq_{\mathcal{ML}(\mathsf{cp},\mathsf{rm})} \mathcal{M}', w'$, then (by Lemma 4.6) $\mathcal{M}_{\mathsf{rm}(\pi)}, w \leftrightarroweq_{\mathcal{ML}(\mathsf{cp},\mathsf{rm})} \mathcal{M}'_{\mathsf{rm}(\pi)}, w'$, and by I.H. $\mathcal{M}'_{\mathsf{rm}(\pi)}, w' \models \varphi$. As a result, $\mathcal{M}', w' \models \mathsf{rm}(\pi)\varphi$.

cp(Q, q)φ: Suppose $\mathcal{M}, w \models \mathsf{cp}(Q,q)\varphi$. Then we have $\mathcal{M}_{\mathsf{cp}(Q)}, (w,q) \models \varphi$. By $\mathcal{M}, w \leftrightarroweq_{\mathcal{ML}(\mathsf{cp},\mathsf{rm})} \mathcal{M}', w'$ and Lemma 4.6, we have $\mathcal{M}_{\mathsf{cp}(Q)}, (w,q) \leftrightarroweq_{\mathcal{ML}(\mathsf{cp},\mathsf{rm})} \mathcal{M}'_{\mathsf{cp}(Q)}, (w',q)$. By I.H. $\mathcal{M}'_{\mathsf{cp}(Q)}, (w',q) \models \varphi$. Therefore, $\mathcal{M}', w' \models \mathsf{cp}(Q,q)\varphi$. □

Definition 4.8. *A language $\mathcal{L}$ has the* tree model property *if for all formula $\varphi \in \mathcal{L}$ satisfiable, φ is satisfied in the root of a tree.*

As a consequence of the tree model property for $\mathcal{ML}(\Box^{-1})$ it immediately follows:

Corollary 4.9. *The language $\mathcal{ML}(\mathsf{cp},\mathsf{rm})$ has the tree model property.*

Notice that for $\mathcal{ML}(\mathsf{cp},\mathsf{rm}^1)$ bisimulations need only satisfy the (zig) and (zag) conditions, as was the case for $\mathcal{ML}$, because removals are restricted to paths of length one.

We already mentioned that $\mathcal{AML}$ has the same expressive power that $\mathcal{ML}$, and it is clear that both are not more expressive than $\mathcal{ML}(\mathsf{cp},\mathsf{rm})$ (as $\mathcal{ML}(\mathsf{cp},\mathsf{rm})$ is a conservative extension of $\mathcal{ML}$). The models shown in Example 4.1 prove that $\mathcal{ML}(\mathsf{cp},\mathsf{rm})$ is actually more expressive than $\mathcal{AML}$.

5. Relation between action models and logics with copy and remove

In this section we investigate the relation between $\mathcal{ML}(\mathsf{cp},\mathsf{rm}^1)$ and action models. First, we introduce an equivalence preserving translation from $\mathcal{AML}$ to $\mathcal{ML}(\mathsf{cp},\mathsf{rm}^1)$. Then we show that action models (with post-conditions) can encode cp and rm operations.

5.1. Embedding action models into $\mathcal{ML}(\mathsf{cp},\mathsf{rm}^1)$

We introduce the translation that formally defines the operations shown in Figure 2.

Definition 5.1. *For all paths π_1, π_2 of length 1, define the shorthand $\mathsf{rm}(\pi_1||\pi_2)\varphi$ for $\mathsf{rm}(\pi_1)\mathsf{rm}(\pi_2)\varphi$. Notice that as π_1 and π_2 are paths of length 1, and given that we are only considering Boolean tests, then $||$ is commutative. Let $\varphi \in \mathcal{AML}$ such that the domain of action models in φ are disjoint. The translation Tr from $\mathcal{AML}$-formulas to $\mathcal{ML}(\mathsf{cp},\mathsf{rm}^1)$-formulas is defined as follows:*

$$
\begin{array}{lcl}
\mathsf{Tr}(p) & = & p \\
\mathsf{Tr}(\neg\varphi) & = & \neg\mathsf{Tr}(\varphi) \\
\mathsf{Tr}(\Diamond_a\varphi) & = & \Diamond_a\mathsf{Tr}(\varphi) \\
\mathsf{Tr}(\varphi\wedge\psi) & = & \mathsf{Tr}(\varphi)\wedge\mathsf{Tr}(\psi) \\
\mathsf{Tr}([\mathcal{E},e_1]\varphi) & = & \mathsf{pre}(e_1)\to\mathsf{cp}(p_{e_1},\ldots,p_{e_n})\mathsf{rm}(\rho)\mathsf{rm}(\sigma)\mathsf{Tr}(\varphi),
\end{array}
$$

where
$\mathcal{E} = \langle E, \rightarrow, \mathsf{pre}\rangle$ *is an action model with* $E = \{e_1, \ldots, e_n\}$ *and*

$$\rho \quad \equiv \bigsqcup_{e_i \in E, a \in \mathsf{AGT}} \top?; a; (p_{e_i} \wedge \neg \mathsf{pre}(e_i))?$$
$$\sigma \quad \equiv \bigsqcup_{e_i, e_j \in E, a \in \mathsf{AGT}} p_{e_i}?; a; p_{e_j}? \qquad \textit{if } e_i \not\rightarrow_a e_j.$$

Cases that do not involve dynamic operations are trivial. Let us analyze the translation of a modality with action models. The translation follows the steps we describe in Figure 2. The antecedent $\mathsf{pre}(e_1)$ is exactly the same clause as for model updates (considering the pointed action model $\mathcal{E}, e_1$ as the desired update). For each action point $e_i \in E$, we consider a propositional symbol p_{e_i}. The $\mathsf{cp}(p_{e_1} \ldots p_{e_n})$ operation replicates the original model as many times as action points in E. This operation generates the cartesian product $W \times E$ and it corresponds to the first **copy** operation in Figure 2. However, the model $\mathcal{M} \otimes \mathcal{E}$ might not result in the whole cartesian product. To cut access to the unwanted part of the model we introduce $\mathsf{rm}(\rho)$. The path expression ρ characterizes all the edges we introduced by the previous $\mathsf{cp}(p_{e_1} \ldots p_{e_n})$ pointing to p_{e_i}-states which do not satisfy the corresponding $\mathsf{pre}(e_i)$. This operation correspond to the second **remove** operation in Figure 2. It remains to restrict the obtained accessibility relation as specified by the action model. This is done by $\mathsf{rm}(\sigma)$. Remember that $((v, d), (u, f)) \in R'_a$ in $\mathcal{M} \otimes \mathcal{E}$ if and only if $(v, u) \in R_a$ and $d \rightarrow_a f$. This operation correspond to the third **remove** operation in Figure 2. $\mathsf{rm}(\sigma)$ deletes all a-edges $((w, p_{e_i}), (w, p_{e_j}))$ such that in the action model there is no a-edge from e_i to e_j, for all $a \in \mathsf{AGT}$.

We obtain a model which is $\mathcal{AML}$-bisimilar to $\mathcal{M} \otimes \mathcal{E}$.

Now we can prove:

Theorem 5.2. *Let* φ *be an* $\mathcal{AML}$*-formula and* $\mathcal{M}, w$ *a pointed model, we have*

$$\mathcal{M}, w \models \varphi \textit{ iff } \mathcal{M}, w \models \mathsf{Tr}(\varphi).$$

Proof. The next claim will be useful in the proof.

Claim 1. *Let* $\mathcal{M} = \langle W, R, V\rangle$ *be a model,* $w \in W$*,* $\mathcal{E} = \langle E, \rightarrow, \mathsf{pre}\rangle$ *be an action model with* $E = \{e_1, \ldots, e_n\}$*, and let* $\mathcal{M}' = ((\mathcal{M}_{\mathsf{cp}(p_{e_1}, \ldots, p_{e_n})})_{\mathsf{rm}(\rho)})_{\mathsf{rm}(\sigma)}$*, with* ρ, σ *as in Definition 5.1. Then* $(\mathcal{M} \otimes \mathcal{E}), (w, e_1) \leftrightarrows^P_{\mathcal{AML}} \mathcal{M}', (w, p_{e_1})$*, where* $P = \{p_{e_1}, \ldots, p_{e_n}\}$*.*

Proof (of the Claim). Let us denote by $\mathcal{M} \otimes \mathcal{E} = \langle W^{\otimes}, R^{\otimes}, V^{\otimes}\rangle$, be the restricted product of $\mathcal{M}$ and $\mathcal{E}$. Define the relation $Z = \{(w, e_1), (w, p_{e_1})) \mid (w, e_1) \in R^{\otimes}_a$ and $(w, p_{e_1}) \in R'_a$, for some $a\}$. We verify the conditions (atomic harmony) with respect to P, (zig) and (zag), in order to prove that Z is a P-bisimulation for $\mathcal{AML}$.

- (atomic harmony). By definition of $\otimes$, we have $(w, e_1) \in V^{\otimes}(p)$ if and only if $w \in V(p)$, if and only if (by $\models$ for cp) $(w, p_{e_1}) \in V_{\mathsf{cp}(\{p_{e_1}, \ldots, p_{e_n}\})}$.

- (zig). Suppose there is (v, e_2) such that $((w, e_1), (v, e_2)) \in R_a^{\otimes}$. By definition of $\otimes$, we have that $(w, v) \in R_a$ and $e_1 \rightarrow_a e_2$.

  By the cp part, we know that if $(w, v) \in R_a$ then $((w, p_{e_1}), (v, p_{e_2})) \in (R_{\mathsf{cp}(\{p_{e_1}, \ldots, p_{e_n}\})})_a$. We need to check that after the two removes, this edge has not been removed.

  ρ removes the edges such that p_{e_i} and $\neg\mathsf{pre}(e_i)$ hold. Because we assume $((w, e_1), (v, e_2)) \in R_a^{\otimes}$, we know that $\mathcal{M}, v \models \mathsf{pre}(e_1)$, then ρ does not remove the edge between (w, p_{e_1}) and (v, p_{e_2}). On the other hand, σ does not remove it because we assume $e_1 \rightarrow_a e_2$. By inductive hypothesis, we have $(v, e_2)Z(v, p_{e_2})$.

- (zag). This condition can be easily checked by applying the same steps as for (zig), but in the other direction. □

The proof of the theorem is by structural induction. Let $\mathcal{M} = \langle W, R, V \rangle$ be a model, $\mathcal{E} = \langle E, \rightarrow, \mathsf{pre} \rangle$ an action model, $w \in W$ and $e_1 \in E$. We will discuss the case involving dynamic operations. We need to prove that $\mathcal{M}, w \models [\mathcal{E}, e_1]\varphi$ iff $\mathcal{M}, w \models \mathsf{Tr}([\mathcal{E}, e_1]\varphi)$. Assume $\mathcal{M}, w \models [\mathcal{E}, e_1]\varphi$, then by definition of $\models$ we have $\mathcal{M}, w \models \mathsf{pre}(e_1) \rightarrow (\mathcal{M} \otimes \mathcal{E}), (w, e_1) \models \varphi$. Let $\mathcal{M}' = ((\mathcal{M}_{\mathsf{cp}(p_{e_1}, \ldots p_{e_n})})_{\mathsf{rm}(\rho)})_{\mathsf{rm}(\sigma)}$.

By Claim 1 $(\mathcal{M} \otimes \mathcal{E}), (w, e_1) \leftrightarrows \mathcal{M}', (w, p_{e_1})$, then (by the invariance theorem for $\mathcal{AML}$) $(\mathcal{M} \otimes \mathcal{E}), (w, e_1) \models \varphi$ if and only if $\mathcal{M}', (w, p_{e_1}) \models \varphi$. Then, by inductive hypothesis $\mathcal{M}', (w, p_{e_1}) \models \mathsf{Tr}(\varphi)$. Hence (by definition of Tr) we have that $\mathcal{M}, w \models \mathsf{Tr}([\mathcal{E}, e_1]\varphi)$. □

We see the encoding above applied to a concrete update in Figure 2, obtaining a model which is bisimilar to the updated model of Figure 1.

5.2. Decomposing $\mathcal{ML}(\mathsf{cp}, \mathsf{rm}^1)$ into action models

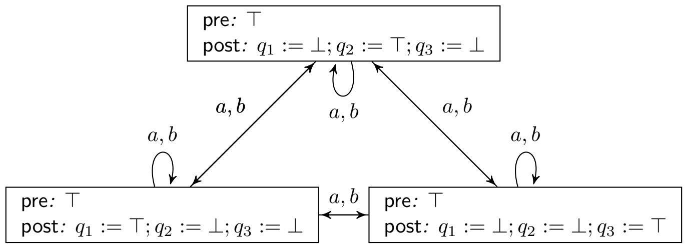
We show now that copy and remove can be seen as action models in $\mathcal{AML}^+$. This is valuable, as it demonstrates that action models can be decompose in simpler action models. This decomposition can be obtained by translating first into $\mathcal{ML}(\mathsf{cp}, \mathsf{rm}^1)$ and then considering copy and remove again as basic action models.

Consider the copy action $\mathsf{cp}(Q)$. The copy operator can be modeled as an action model $\mathcal{E}(\mathsf{cp}(Q)) = \langle Q, \rightarrow, \mathsf{pre}, \mathsf{post} \rangle$ such that $\rightarrow_a = Q \times Q$ (for all $a \in \mathsf{AGT}$), and for all $q \in Q$:

$$\begin{array}{lcllcll} \mathsf{pre}(q) & = & \top & \mathsf{post}(q)(p) & = & \bot & \text{for } p \in Q \setminus \{q\} \\ \mathsf{post}(q)(q) & = & \top & \mathsf{post}(q)(p) & = & p & \text{for } p \in \mathsf{PROP} \setminus Q. \end{array}$$

We note that for all $r \in \mathsf{PROP} \setminus Q$ the value is not affected at the execution of this action, as the finite subset of propositional symbols that is assigned a post-condition is the set Q.

Example 5.3. *The action model $\mathcal{E}(\mathsf{cp}(q_1, q_2, q_3))$ for two agents a and b is depicted below:*



Consider the translation $' : \mathcal{ML}(\mathsf{cp}) \to \mathcal{AML}$ such that

$$(\mathsf{cp}(Q, q)\varphi)' = [\mathcal{E}(\mathsf{cp}(Q)), q]\varphi',$$

and commutes with all other operators.

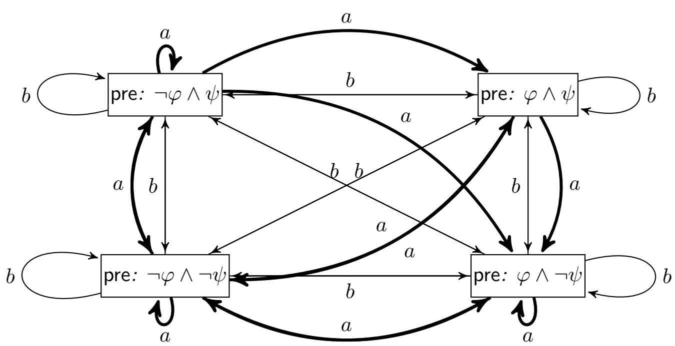
Proposition 5.4. *Let $\mathcal{M}, w$ be a pointed model, and let φ be a $\mathcal{ML}(\mathsf{cp})$-formula. Then $\mathcal{M}, w \models \varphi$ iff $\mathcal{M}, w \models \varphi'$.*

The action model $\mathcal{E}(\mathsf{rm}(\varphi?; a; \psi?)) = \langle E, \to, \mathsf{pre}\rangle$ is defined as

$$\begin{array}{lll} E & = \{00, 10, 01, 11\} & \mathsf{pre}(00) = \neg\varphi \wedge \neg\psi, \\ \to_a & = (E \times E) \setminus \{(10, 01), (10, 11), (11, 01), (11, 11)\} & \mathsf{pre}(10) = \varphi \wedge \neg\psi, \\ \to_b & = (E \times E) \quad \text{for all } b \neq a & \mathsf{pre}(01) = \neg\varphi \wedge \psi, \\ & & \mathsf{pre}(11) = \varphi \wedge \psi. \end{array}$$

This action model corresponds to the operation of removing all a-edges (w, v) such that φ holds in w and ψ holds in v.

Example 5.5. *Suppose we have two agents a and b. Then $\mathcal{E}(\mathsf{rm}(\varphi?; a; \psi?))$ is depicted below:*



Consider the translation $'' : \mathcal{ML}(\mathsf{rm}^1) \to \mathcal{AML}$ such that

$$(\mathsf{rm}(\varphi?; a; \psi?)\theta)'' = [\mathcal{R}, 00 \cup \mathcal{R}, 01 \cup \mathcal{R}, 10 \cup \mathcal{R}, 11]\theta''$$

where $\mathcal{R} = \mathcal{E}(\mathsf{rm}(\varphi?; a; \psi?))$ and where $[\alpha \cup \alpha']\phi$ is a shorthand for $[\alpha]\varphi \wedge [\alpha']\varphi$, and commutes with all other operators.

Proposition 5.6. *Let $\mathcal{M}, w$ be a pointed model, and let φ be a $\mathcal{ML}(\mathsf{rm}^1)$-formula. Then $\mathcal{M}, w \models \varphi$ iff $\mathcal{M}, w \models \varphi''$.*

Combining the previous results we obtain that every finite action model has the same update effect as the composition of a copy action model and two remove action models, where we use the translation Tr from Definition 5.1 but replace the copy and remove actions there by their correspondent action model.

6. Complexity of deciding satisfiability

In this section, we will focus on the satisfiability problem restricted to fragments, adapting results obtained in [5].

6.1. Complexity of the fragment $\mathcal{ML}(\mathsf{cp})$

Let Σ be an arbitrary set of pairs (Q, q) where $Q \subseteq \mathsf{PROP}$ and $q \in Q$. We define the translation Tr_Σ that maps formulas $\varphi \in \mathcal{ML}(\mathsf{cp})$ to formulas of $\mathcal{ML}$ as follows:

$$\begin{array}{lcl}
\mathsf{Tr}_\Sigma(p) & = & p \\
\mathsf{Tr}_\Sigma(\neg\varphi) & = & \neg\mathsf{Tr}_\Sigma(\varphi) \\
\mathsf{Tr}_\Sigma(\varphi \wedge \psi) & = & \mathsf{Tr}_\Sigma(\varphi) \wedge \mathsf{Tr}_\Sigma(\psi) \\
\mathsf{Tr}_\Sigma(\Box_a\varphi) & = & \Box_a(\bigwedge_{(Q_i,p_i)\in\Sigma} p_i \rightarrow \mathsf{Tr}_\Sigma(\varphi)) \\
\mathsf{Tr}_\Sigma(\mathsf{cp}(Q,q)\varphi) & = & \mathsf{Tr}_{\Sigma\setminus\{(Q,q)\}}(\varphi).
\end{array}$$

We define by induction:

$$\begin{array}{lcl}
\mathcal{M}_{\mathsf{cp}(\{(Q,q)\}\cup\Sigma)} & := & \left(\mathcal{M}_{\mathsf{cp}(\Sigma)}\right)_{\mathsf{cp}(Q)} \\
\mathcal{M}_{\mathsf{cp}(\emptyset)} & := & \mathcal{M}.
\end{array}$$

For our purposes, the definition of $\mathcal{M}_\Sigma$ does not depend on the order in Σ.

Proposition 6.1. *For all $\varphi \in \mathcal{ML}(\mathsf{cp})$, let $\Sigma(\varphi)$ the set of all pairs (Q, q) appearing in copy operators in φ. Then*

$$\mathcal{M}, w \models \varphi \quad \textit{iff} \quad \mathcal{M}_{\mathsf{cp}(\Sigma(\varphi))}, w_{\Sigma(\varphi)} \models \mathsf{Tr}_{\Sigma(\varphi)}(\varphi),$$

where $w_{\Sigma(\varphi)}$ is the corresponding evaluation point after $|\,\Sigma(\varphi)\,|$ consecutive cp *operations are applied from the point w.*

PROOF. The proof is by induction on the structure of $\varphi \in \mathcal{ML}(\mathsf{cp})$.

$\boldsymbol{\varphi = p}$ **:** $\mathcal{M}, w \models p$ if and only if $\mathcal{M}_{\mathsf{cp}(\emptyset)}, w \models \mathsf{Tr}_\emptyset(p)$ (by definition of $\mathcal{M}_{\mathsf{cp}(\emptyset)}$ and Tr) if and only if (by definition of $\Sigma(\varphi)$) $\mathcal{M}_{\mathsf{cp}(\Sigma(p))}, w \models \mathsf{Tr}_{\Sigma(p)}(p)$

$\boldsymbol{\varphi = \neg\psi}$ and $\boldsymbol{\varphi = \psi \wedge \chi}$ **:** Are trivial by inductive hypothesis.

$\varphi = \Box_a\psi$ **:** By definition of $\models$, $\mathcal{M}, w \models \Box_a\psi$ iff for all v such that $(w,v) \in R_a$, $\mathcal{M}, v \models \psi$. $\Sigma(\Box_a\psi) = \Sigma(\psi)$. By inductive hypothesis, we have $\mathcal{M}_{\mathsf{cp}(\Sigma(\psi))}, v_{\Sigma(\psi)} \models \mathsf{Tr}_{\Sigma(\psi)}(\psi)$. This is the same as $\mathcal{M}_{\mathsf{cp}(\Sigma(\Box_a\psi))}, v_{\Sigma(\Box_a\psi)} \models \mathsf{Tr}_{\Sigma(\Box_a\psi)}(\psi)$. But (because $v_{\Sigma(\Box_a\psi)}$ is an arbitrary copy of the successors of w) $\mathcal{M}_{\mathsf{cp}(\Sigma(\Box_a\psi))}, w_{\Sigma(\Box_a\psi)} \models \Box_a(\bigwedge_{(Q_i,p_i)\in\Sigma(\Box_a\psi)} p_i \rightarrow \mathsf{Tr}_\Sigma(\psi))$, if and only if $\mathcal{M}_{\mathsf{cp}(\Sigma)}, w_{\Sigma(\Box_a\psi)} \models \mathsf{Tr}_{\Sigma(\Box_a\psi)}(\Box_a\psi)$.

$\varphi = \mathsf{cp}(Q,p)\psi$ **:** $\mathcal{M}, w \models \mathsf{cp}(Q,p)\psi$ iff $\mathcal{M}_{\mathsf{cp}(Q)}, (w,p) \models \psi$. By inductive hypothesis, $(\mathcal{M}_{\mathsf{cp}(Q)})_{\Sigma(\psi)}, w_{\Sigma(\varphi)} \models \mathsf{Tr}_{\Sigma(\psi)}(\psi)$. Because the definition of $\mathcal{M}_{\mathsf{cp}(\Sigma)}$ (which does not depend on the order of Σ), and for all q, $(Q,q) \notin \Sigma(\psi)$ (because in each occurrence of $\mathsf{cp}(Q,q)$, Q is fresh), we have that $\mathcal{M}_{\mathsf{cp}(\Sigma(\psi)\cup\{(Q,p)\})}, w_{\Sigma(\varphi)} \models \mathsf{Tr}_{\Sigma(\psi)\setminus\{(Q,p)\}}(\psi)$. Hence, by definition we have $\mathcal{M}_{\mathsf{cp}(\Sigma(\psi)\cup\{(Q,p)\})}, w_{\Sigma(\varphi)} \models \mathsf{Tr}_{\Sigma(\psi)\cup\{(Q,p)\}}(\mathsf{cp}(Q)\psi)$, which is the same that $\mathcal{M}_{\mathsf{cp}(\Sigma(\mathsf{cp}(Q,p)\psi))}, w_{\Sigma(\varphi)} \models \mathsf{Tr}_{\Sigma(\mathsf{cp}(Q)\psi)}(\mathsf{cp}(Q)\psi)$. $\Box$

We will show the upper bound for the class PSpace, by providing a tableau-based algorithm which uses polynomial space. Notice that the algorithm takes as argument an $\mathcal{ML}$-formula, a set of sequences of propositional symbols and a set of formulas. At the end, we use the previous result to complete the proof.

Proposition 6.2. *The following problem is in* PSpace*:*

- *input: a formula* $\varphi \in \mathcal{ML}$*;* Σ *a set of pairs* (Q_i, q_i)*, such that* $Q_i \subseteq \mathsf{PROP}$ *and* $q_i \in Q_i$*;*
- *output: yes iff there exists a model* $\mathcal{M}$ *such that* $\mathcal{M}_{\mathsf{cp}(\Sigma)}, w_\Sigma \models \varphi$*.*

Proof. For this proof we take $\Diamond_a$ as the fundamental operator and consider $\Box_a$ as being defined as $\Box_a\varphi \leftrightarrow \neg\Diamond_a\neg\varphi$. We adapt the standard tableau method for $\mathcal{ML}$ (see [16]) in order to obtain a PSpace procedure for our problem, shown in Algorithm 1. The set ν is called a *modal valuation* over a set of formulas Γ if and only if $\nu \subseteq \mathsf{PROP} \cup \{\psi \mid \psi \in \mathsf{Diam}(\Gamma)\}$, where $\mathsf{Diam}(\Gamma) = \bigcup_{\varphi\in\Gamma} \mathsf{Diam}(\varphi)$, and $\mathsf{Diam}(\varphi)$ is defined inductively as follows:

$$\begin{array}{lcl}
\mathsf{Diam}(p) & = & \emptyset \\
\mathsf{Diam}(\varphi \wedge \varphi') & = & \mathsf{Diam}(\varphi) \cup \mathsf{Diam}(\varphi') \\
\mathsf{Diam}(\Diamond_a\varphi) & = & \{\Diamond_a\varphi\} \\
\mathsf{Diam}(\neg\Diamond_a\varphi) & = & \{\Diamond_a\varphi\}.
\end{array}$$

We define the relation $\models$ to say that a valuation satisfies a formula as follows:

$$\begin{array}{lll}
\nu \models p & \text{iff} & p \in \nu \\
\nu \models \Diamond_a\varphi & \text{iff} & \Diamond_a\varphi \in \nu \\
\nu \models \neg\varphi & \text{iff} & \nu \not\models \varphi \\
\nu \models \varphi \wedge \varphi' & \text{iff} & \nu \models \varphi \text{ and } \nu \models \varphi'.
\end{array}$$

A valuation c is called a *copy valuation* for a set Σ of pairs (Q_i, q_i), if and only if c contains exactly one propositional symbol in each Q_i. Notice that, given a modal valuation ν and a copy valuation c, $\nu \cup c$ is a modal valuation.

Algorithm 1 Satisfiability for the fragment $\mathcal{ML}(\mathsf{cp})$

procedure $\mathsf{SAT}(\varphi, \Gamma, \Sigma)$
  choose some modal valuation ν over $\Gamma \cup \{\varphi\}$
  for all copy valuation c over Σ **do**
    if $\nu \cup c \not\models \bigwedge_{\gamma_i \in \Gamma} \gamma_i$ **then**
      return UNSAT
    end if
  end for
  if for no copy valuation c we have $\nu \cup c \models \varphi \wedge \bigwedge_{\gamma_i \in \Gamma} \gamma_i$ **then**
    return UNSAT
  end if
  for all $\Diamond_a \psi \in \nu$ **do**
    if $\mathsf{SAT}(\psi, \{\neg\chi \mid \neg\Diamond_a \chi \in \nu\}, \Sigma) =$UNSAT **then**
      return UNSAT
    end if
  end for
  return SAT
end procedure

The procedure takes three arguments: a formula φ, a set of formulas Γ and a set Σ which contains pairs $(Q_i, q_i) \in 2^{\mathsf{PROP}} \times \mathsf{PROP}$. The set Γ ais used to abstract subformulas of φ. Modal valuations treat formulas as propositional symbols. The algorithm is the standard tableau algorithm used to check satisfiability for $\mathcal{ML}$, adapted to manage a set Σ, which represents possible copies of a model. As for $\mathcal{ML}$, the algorithm takes only polynomial space.

Theorem 6.3. *The satisfiability problem of* $\mathcal{ML}(\mathsf{cp})$ *is* PSPACE*-complete.*

PROOF. PSPACE-hardness follows from PSPACE-completeness of the satisfiability problem for $\mathcal{ML}$. In order to prove completeness, we can use Proposition 6.2, and test whether there exists a model $\mathcal{M}$ such that $\mathcal{M}_{\Sigma(\varphi)}, w_{\Sigma(\varphi)} \models \mathsf{Tr}_{\Sigma(\varphi)}(\varphi)$. This can be done (by Proposition 6.1) by invoking

$$\mathsf{SAT}(\mathsf{Tr}_{\Sigma(\varphi)}(\varphi) \wedge \textstyle\bigwedge_{(Q_i, p_i) \in \Sigma(\varphi)} p_i, \emptyset, \Sigma(\varphi)).$$

6.2. Complexity of the fragment $\mathcal{ML}(\mathsf{rm})$

We will show that $\mathcal{ML}(\mathsf{rm})$ can be translated into $\mathcal{ML}(\Box^{-1})$, the basic modal logic $\mathcal{ML}$ extended with $\Box^{-1}$. As we have mentioned, all remove operations can be transformed in the normal form $\mathsf{rm}(\varphi_1?; a_1; \ldots; a_{n-1}; \varphi_n?)\psi$. We introduce reduction axioms to get an $\mathcal{ML}(\Box^{-1})$-formula, and prove that any $\mathcal{ML}(\mathsf{rm})$-formula, is equivalent to an $\mathcal{ML}(\Box^{-1})$-formula.

First, define the abbreviations $\Diamond_{i,j}$, $\Diamond^{-1}_{i,j}$, for a fix path expression $\pi =$

$\varphi_1?; a_1; \ldots; a_{n-1}; \varphi_n?$:

$$\Diamond_{i,j} = \begin{cases} \top & j < i \\ \Diamond_{a_i}\varphi_{i+1} & i = j \\ \Diamond_{a_i}(\varphi_{i+1} \wedge \Diamond_{i+1,j}) & i < j \end{cases} \qquad \Diamond^{-1}_{i,j} = \begin{cases} \top & j < i \\ \Diamond^{-1}_{a_i}\varphi_i & i = j \\ \Diamond^{-1}_{a_j}(\Diamond^{-1}_{i,j-1} \wedge \varphi_j) & i < j \end{cases}$$

Now define $rm_i^\pi = \Diamond^{-1}_{1,i-1} \wedge \varphi_i \wedge \Diamond_{i,n-1}$. Informally rm_i^π means "the current state is at position i in a path that matches $\pi = \varphi_1?; a_1; \varphi_2?; a_2; \ldots; a_{n-1}; \varphi_n?$ which is going to be deleted". For instance, rm_i^π, $1 \leq i \leq n$ are defined as:

$$\begin{array}{lcl} rm_1^\pi & = & \varphi_1 \wedge (\Diamond_{a_1}\varphi_2 \wedge (\Diamond_{a_2}\varphi_3 \ldots \wedge \Diamond_{a_{n-2}}(\varphi_{n-1} \wedge \Diamond_{a_{n-1}}\varphi_n)\ldots)) \\ rm_2^\pi & = & \Diamond^{-1}_{a_1}\varphi_1 \wedge \varphi_2 \wedge (\Diamond_{a_2}\varphi_3 \ldots \wedge \Diamond_{a_{n-2}}(\varphi_{n-1} \wedge \Diamond_{a_{n-1}}\varphi_n)\ldots) \\ & \ldots & \\ rm_{n-1}^\pi & = & \Diamond^{-1}_{a_{n-2}}(\Diamond^{-1}_{a_{n-3}}(\ldots(\Diamond^{-1}_{a_1}\varphi_1 \wedge \varphi_2) \wedge \varphi_3)\ldots) \wedge \varphi_{n-1} \wedge \Diamond_{a_{n-1}}\varphi_n \\ rm_n^\pi & = & \Diamond^{-1}_{a_{n-1}}(\Diamond^{-1}_{a_{n-2}}(\ldots(\Diamond^{-1}_{a_1}\varphi_1 \wedge \varphi_2) \wedge \varphi_3 \ldots) \wedge \varphi_{n-1}) \wedge \varphi_n. \end{array}$$

Lemma 6.4. *Let $\mathcal{M}=\langle W, R, V\rangle$ be a model, $w \in W$ and $\pi=\varphi_1?; a_1; \varphi_2?; \ldots; \varphi_n?$ a path expression. Let i be such that $0 \leq i \leq n$, then*

$$\mathcal{M}, w \models rm_i^\pi \textit{ iff there is some } P \in \mathcal{P}_\pi^\mathcal{M} \textit{ s.t. } P = w_1 a_1 w_2 \ldots w_n,\ w_i = w$$

and for all $w_j \in P$ we have $\mathcal{M}, w_j \models \varphi_j$.

Proof. The proof is by induction on the length of π:

$\boldsymbol{\pi = \varphi_1?}$: $\mathcal{M}, w \models rm_1^\pi$ if and only if $\mathcal{M}, w \models \varphi_1$ (by definition of rm_i^π). But $\mathcal{P}^\mathcal{M}_{\varphi_1?} = \{v \mid \mathcal{M}, v \models \varphi_1\}$ (all the paths are singletons satisfying φ_1), then $w \in \mathcal{P}^\mathcal{M}_{\varphi_1?}$.

$\boldsymbol{\pi = \varphi_1?; a_1; \varphi_2?; \ldots; \varphi_n?}$: Suppose $\mathcal{M}, w \models rm_i^\pi$. By definition of rm^π, we have $\mathcal{M}, w \models \Diamond^{-1}_{1,i-1} \wedge \varphi_i \wedge \Diamond_{i,n-1}$. Now, we know:

1. $\mathcal{M}, w \models \varphi_i$.
2. $\mathcal{M}, w \models \Diamond^{-1}_{1,i-1}$, then by definition of $\Diamond^{-1}_{i,j}$ we have $\mathcal{M}, w \models \Diamond^{-1}_{a_{i-1}}(\Diamond^{-1}_{1,i-2} \wedge \varphi_{i-1})$. By definition of $\models$, there is some $v \in W$ such that $(v, w) \in R_{a_{i-1}}$ and $\mathcal{M}, v \models \Diamond^{-1}_{1,i-2} \wedge \varphi_{i-1}$. Let us define $\pi_1 = \varphi_1?; a_1; \varphi_2?; \ldots; \varphi_{i-1}?$. Then, by definition of rm_i^π, we have $\mathcal{M}, v \models rm_{i-1}^{\pi_1}$, and by I.H., there is a path $P_1 \in \mathcal{P}^\mathcal{M}_{\pi_1}$ such that $P_1 = w_1 a_1 \ldots w_{i-1}$, with $w_{i-1} = v$ and for all $w_j \in P_1$, $\mathcal{M}, w_j \models \varphi_j$ $(0 \leq j \leq i-1)$.
3. $\mathcal{M}, w \models \Diamond_{i,n-1}$, then by definition of $\Diamond_{i,j}$ we have $\mathcal{M}, w \models \Diamond_{a_i}(\varphi_{i+1} \wedge \Diamond_{i+1,n-1})$. By definition of $\models$, there is some $t \in W$ such that $(w, t) \in R_{a_i}$ and $\mathcal{M}, t \models \varphi_{i+1} \wedge \Diamond_{i+1,n-1}$. Let us define $\pi_2 = \varphi_{i+1}?; a_{i+1}; \ldots; \varphi_n?$. Then, by definition of rm_i^π, we have $\mathcal{M}, t \models rm_{i+1}^{\pi_2}$, and by I.H., there is a path $P_2 \in \mathcal{P}^\mathcal{M}_{\pi_2}$ such that $P_2 = w_{i+1} a_{i+1} \ldots w_n$, with $w_{i+1} = t$ and for all $w_j \in P_2$, $\mathcal{M}, w_j \models \varphi_j$ $(i+1 \leq j \leq n)$.

Notice that $\pi = \pi_1; a_{i-1}; \varphi_i?; a_i; \pi_2$. It remains to choose $P{=}P_1 a_{i-1} w_i a_i P_2$ and we have what we wanted. □

We introduce reduction axioms which transform $\mathcal{ML}(\mathsf{rm},\Box^{-1})$-formulas[1] into $\mathcal{ML}(\Box^{-1})$-formulas. We need to define axioms for $\mathcal{ML}(\mathsf{rm},\Box^{-1})$ in order to manage intermediate steps.

Proposition 6.5. *Let* $\pi{=}\varphi_1?;a_1;\varphi_2?;\ldots;\varphi_n?$, $\varphi{=}\mathsf{rm}(\pi)\theta$ *be an* $\mathcal{ML}(\mathsf{rm},\Box^{-1})$*-formula, then the following reduction axioms are valid (we assume that* $\Diamond_a\psi$ *is written as* $\neg\Box_a\neg\psi$*, and similarly for* $\Diamond^{-1}$*).*

$$
\begin{array}{lllll}
(1) & \mathsf{rm}(\pi)p & \leftrightarrow & p, \quad p \in \mathsf{PROP} \\
(2) & \mathsf{rm}(\pi)\neg\psi & \leftrightarrow & \neg\mathsf{rm}(\pi)\psi \\
(3) & \mathsf{rm}(\pi)(\psi\wedge\psi') & \leftrightarrow & (\mathsf{rm}(\pi)\psi \wedge \mathsf{rm}(\pi)\psi') \\
(4) & \mathsf{rm}(\pi)\Box_a\psi & \leftrightarrow & \Box_a\mathsf{rm}(\pi)\psi, \quad \textit{if } a\notin\pi \\
(5) & \mathsf{rm}(\pi)\Box_a^{-1}\psi & \leftrightarrow & \Box_a^{-1}\mathsf{rm}(\pi)\psi, \quad \textit{if } a\notin\pi \\
(6) & \mathsf{rm}(\pi)\Box_a\psi & \leftrightarrow & (\bigwedge_{i\in\{1,\ldots,n-1\ \mid\ a_i=a\}}\neg rm_i^\pi \to \Box_{a_i}\mathsf{rm}(\pi)\psi)\wedge \\
& & & (\bigwedge_{i\in\{1,\ldots,n-1\ \mid\ a_i=a\}}(rm_i^\pi \to \Box_{a_i}(rm_{i+1}^\pi \vee \mathsf{rm}(\pi)\psi))) \\
(7) & \mathsf{rm}(\pi)\Box_a^{-1}\varphi & \leftrightarrow & (\bigwedge_{i\in\{1,\ldots,n-1\ \mid\ a_i=a\}}\neg rm_i^\pi \to \Box_{a_i}^{-1}\mathsf{rm}(\pi)\psi)\wedge \\
& & & (\bigwedge_{i\in\{1,\ldots,n-1\ \mid\ a_i=a\}}(rm_i^\pi \to \Box_{a_i}^{-1}(rm_{i-1}^\pi \vee \mathsf{rm}(\pi)\psi))).
\end{array}
$$

Proof. We prove each of them separately:

1. Suppose $\mathcal{M},w\models\mathsf{rm}(\pi)p$. By definition of $\models$, we have $\mathcal{M}_{\mathsf{rm}(\pi)},w\models p$. Because $\mathsf{rm}(\pi)$ keeps the same valuation in the updated model, $w\in V(p)$. Then (by $\models$), $\mathcal{M},w\models p$.

2. Follows from the self-duality of rm, which is trivial given that it is a global operator.

3. Suppose $\mathcal{M},w\models\mathsf{rm}(\pi)(\psi\wedge\psi')$. Then, by definition of $\models$, $\mathcal{M}_{\mathsf{rm}(\pi)},w\models(\psi\wedge\psi')$, which means $\mathcal{M}_{\mathsf{rm}(\pi)},w\models\psi$ and $\mathcal{M}_{\mathsf{rm}(\pi)},w\models\psi'$. Applying again definition of $\models$, we have $\mathcal{M},w\models\mathsf{rm}(\pi)\psi$ and $\mathcal{M},w\models\mathsf{rm}(\pi)\psi'$, iff $\mathcal{M},w\models\mathsf{rm}(\pi)\psi\wedge\mathsf{rm}(\pi)\psi'$.

4. (5 is straightforward). Suppose $\mathcal{M},w\models\mathsf{rm}(\pi)\Box_{a_i}\psi$. Applying definition of $\models$ twice, we have that for all v such that $(w,v)\in(R_{\mathsf{rm}(\pi)})_{a_i}$, $\mathcal{M}_{\mathsf{rm}(\pi)},v\models\psi$. We assume $a_i\notin\pi$, then $(w,v)\in(R_{\mathsf{rm}(\pi)})_{a_i}$ iff $(w,v)\in R_{a_i}$, then we have for all v such that $(w,v)\in R_{a_i}$, $\mathcal{M}_{\mathsf{rm}(\pi)},v\models\psi$, iff for all v such that $(w,v)\in R_{a_i}$, $\mathcal{M},v\models\mathsf{rm}(\pi)\psi$. Hence by $\models$, $\mathcal{M},w\models\Box_{a_i}\mathsf{rm}(\pi)\psi$.

6. (7 is straightforward). Let $\mathcal{M}=\langle W,R,V\rangle$ be a model, $w\in W$, and let $\mathsf{rm}(\pi)\Box_{a_i}\psi$ be an $\mathcal{ML}(\mathsf{rm},\Box^{-1})$-formula with $\pi=\varphi_1?;a_1;\varphi_2?;\ldots;\varphi_n?$, such that $a_i\in\pi$. We want to prove

$$\mathcal{M},w\models\mathsf{rm}(\pi)\Box_{a_i}\psi \text{ iff } \mathcal{M},w\models\delta\wedge\delta'$$

where

$$
\begin{array}{lll}
\delta & = & \bigwedge_{k\in\{1,\ldots,n-1\ \mid\ a_k=a_i\}}\neg rm_k^\pi \to \Box_{a_k}\mathsf{rm}(\pi)\psi \\
\delta' & = & \bigwedge_{k\in\{1,\ldots,n-1\ \mid\ a_k=a_i\}}(rm_k^\pi \to \Box_{a_k}(rm_{k+1}^\pi \vee \mathsf{rm}(\pi)\psi)).
\end{array}
$$

[1] Let $\mathcal{ML}(\mathsf{rm},\Box^{-1})$ be the fragment $\mathcal{ML}(\mathsf{rm})$ extended with the past operator $\Box^{-1}$.

Let us suppose that $\mathcal{M}, w \models \mathsf{rm}(\pi)\Box_{a_i}\psi$. Then, by definition of $\models$, we have that for all $v \in W$ such that $(w, v) \in (R_{\mathsf{rm}(\pi)})_{a_i}$, $\mathcal{M}_{\mathsf{rm}(\pi)}, v \models \psi$. We will check the two conjuncts δ and δ' separately (for the other direction of the iff, we can assume the two conjuncts together and use the same steps):
1. Suppose $\mathcal{M}, w \models \bigwedge_{k\in\{1,\ldots,n-1 \mid a_k=a_i\}} \neg rm^\pi_k$. By definition of $\models$, we have $\mathcal{M}, w \not\models \bigvee_{k\in\{1,\ldots,n-1 \mid a_k=a_i\}} rm^\pi_k$. It means that there is no $P \in \mathcal{P}^{\mathcal{M}}_\pi$ satisfying Lemma 6.4, such that $w \in P$, hence no deletions have been done traversing w. Then for all $v \in W$, $(w, v) \in R_{a_i}$ iff $(w, v) \in (R_{\mathsf{rm}(\pi)})_{a_i}$. Because we have for all $v \in W$ such that $(w, v) \in (R_{\mathsf{rm}(\pi)})_{a_i}$, $\mathcal{M}_{\mathsf{rm}(\pi)}, v \models \psi$, then for all $v \in W$ such that $(w, v) \in R_{a_i}$, $\mathcal{M}_{\mathsf{rm}(\pi)}, v \models \psi$. Therefore, we have for all $v \in W$ such that $(w, v) \in R_{a_i}$, $\mathcal{M}, v \models \mathsf{rm}(\pi)\psi$, then (by $\models$) $\mathcal{M}, w \models \Box_{a_i}\mathsf{rm}(\pi)\psi$.
2. Suppose now for some arbitrary k, $\mathcal{M}, w \models rm^\pi_k$, where $k \in \{1, \ldots, n - 1 \mid a_k = a_i\}$. By Lemma 6.4 it means that there is a path traversing w that has been deleted. We also know $\mathcal{M}_{\mathsf{rm}(\pi)}, w \models \Box_{a_k}\psi$ by assumption and $k = i$, then for all $v \in W$ such that $(w, v) \in (R_{\mathsf{rm}(\pi)})_{a_k}$, $\mathcal{M}_{\mathsf{rm}(\pi)}, v \models \psi$. Then, for all $u \in W$ such that $(w, u) \in R_{a_k}$, either $\mathcal{M}_{\mathsf{rm}(\pi)}, u \models \psi$ or $u \in P$, with $P \in \mathcal{P}^{\mathcal{M}}_\pi$, and u is at position $k + 1$ (because w is at position $k = i$), i.e., $\mathcal{M}, u \models rm^\pi_{k+1}$ (by Lemma 6.4). Therefore, $\mathcal{M}, w \models \Box_{a_k}(rm^\pi_{k+1} \vee \mathsf{rm}(\pi)\psi)$. □

Proposition 6.5 provides a way to eliminate all the rm operators in a formula. We can design an algorithm to transform any $\mathcal{ML}(\mathsf{rm}, \Box^{-1})$-formula into a $\mathcal{ML}(\Box^{-1})$-formula, which applies exhaustively the reduction axioms. Notice that each reduction axiom reduces the formula on the left to an $\mathcal{ML}(\mathsf{rm}, \Box^{-1})$-formula on the right, where the rm operation is moved inwards till it can finally be eliminated.

Proposition 6.6. *Let $\mathcal{M}, w$ be a pointed model. For all $\varphi \in \mathcal{ML}(\mathsf{rm}, \Box^{-1})$, there exists $\varphi' \in \mathcal{ML}(\Box^{-1})$ such that $\mathcal{M}, w \models \varphi$ iff $\mathcal{M}, w \models \varphi'$.*

The next theorem now follows.

Theorem 6.7. *The satisfiability problem for $\mathcal{ML}(\mathsf{rm})$ is decidable.*

The reduction axioms that eliminate $\mathsf{rm}(\pi)$ produce an exponential blow up in the size of the formula. If we consider only formulas in $\mathcal{ML}(\mathsf{rm}^1)$, we can avoid the exponential blow up.

Proposition 6.8. *Let $\mathcal{M} = \langle W, R, V\rangle$ be a model, θ, φ and ψ be $\mathcal{ML}(\mathsf{rm}^1)$-formulas and $a \in \mathsf{AGT}$. Then*

$$\mathcal{M}, w \models \mathsf{rm}(\varphi?; a; \psi?)\Box_a\theta \text{ iff } \mathcal{M}, w \models \Box_a((\psi \wedge \Diamond^{-1}\varphi) \vee \mathsf{rm}(\varphi?; a; \psi?)\theta).$$

Proof. Let us suppose that $\mathcal{M}, w \models \mathsf{rm}(\varphi?; a; \psi?)\Box_a\theta$. Then, we have that for all $v \in W$ s.t. $(w, v) \in (R_{\mathsf{rm}(\varphi?;a;\psi?)})_a$, $\mathcal{M}_{\mathsf{rm}(\varphi?;a;\psi?)}, w \models \theta \quad \otimes$. Let u be s.t. $(w, u) \in R_a$, and let suppose $\mathcal{M}, u \models \neg(\psi \wedge \Diamond^{-1}\varphi)$. This means that $(w, u) \in R_a$ iff $(w, u) \in (R_{\mathsf{rm}(\varphi?;a;\psi?)})_a$. Then (by $\otimes$) $\mathcal{M}_{\mathsf{rm}(\varphi?;a;\psi?)}, u \models \theta$ iff (by $\models$) $\mathcal{M}, u \models \mathsf{rm}(\varphi?; a; \psi?)\theta$, iff $\mathcal{M}, w \models \Box_a(\neg(\psi \wedge \Diamond^{-1}_a\varphi) \to \mathsf{rm}(\varphi?; a; \psi?)\theta)$. □

Then, next theorem immediately follows:

Theorem 6.9. *The satisfiability problem for $\mathcal{ML}(\mathsf{rm}^1)$ is* PSpace*-complete.*

6.3. Complexity of the fragment $\mathcal{ML}(\mathsf{cp}, \mathsf{rm}^1)$

We showed that there is a polynomial translation from $\mathcal{ML}(\mathsf{cp}, \mathsf{rm}^1)$ into a dynamic epistemic modal logic with action models with both pre-conditions and post-conditions, that preserves satisfiability. In [7], it is proved that the satisfiability problem for dynamic epistemic modal logic with action models with pre-conditions and *without* post-conditions is in NExpTime. We can handle post-conditions in NExpTime adapting the tableau method of [7][2].

Theorem 6.10. *The satisfiability problem for $\mathcal{AML}^+$ is in* NExpTime.

Proof. We will provide a tableau method adapted from the case without post-conditions [7]. Let LAB be a countable set of labels designed to represent states of the model $(\mathcal{M}, w)$ we are trying to construct. Our tableau method manipulates terms that we call *tableau terms* and they are of the following kind:

- $(\sigma \quad \mathcal{E}_1, e_1; \dots; \mathcal{E}_i, e_i \quad \varphi)$ where $\sigma \in \mathsf{LAB}$ is a symbol (that represents a state in the initial model) and for all $j \in \{1, \dots, i\}$, $\mathcal{E}_j, e_j$ is an action model. This term means that φ is true in the state denoted by σ after the execution of the sequence $\mathcal{E}_1, e_1, \dots, \mathcal{E}_i, e_i$ and that the sequence is executable in the state denoted by σ;
- $(\sigma \quad \mathcal{E}_1, e_1; \dots; \mathcal{E}_i, e_i \quad \checkmark)$ means that the sequence $\mathcal{E}_1, e_1, \dots, \mathcal{E}_i, e_i$ is executable in the state denoted by σ. The symbol $\checkmark$ means that the state survives a sequence of pointed action models;
- $(\sigma \quad \mathcal{E}_1, e_1; \dots; \mathcal{E}_i, e_i \quad \otimes)$ means that the sequence $\mathcal{E}_1, e_1, \dots, \mathcal{E}_i, e_i$ is not executable in the state denoted by σ. The symbol $\otimes$ means that the state does not survive the sequence of pointed action models.
- $(\sigma R_a \sigma_1)$ means that the state denoted by σ is linked by R_a to the state denoted by σ_1;
- $\bot$ denotes an inconsistency.

A *tableau rule* is represented by a *numerator* $\mathcal{N}$ above a line and a finite list of *denominators* $\mathcal{D}_1, \dots, \mathcal{D}_k$ below this line, separated by vertical bars. In the following σ denotes a symbol for states. Σ, Σ', etc., denote sequences of pointed action models. ϵ denotes the empty sequence of pointed action models.

A *tableau tree* is a finite tree with a set of tableau terms at each node. A rule with numerator $\mathcal{N}$ is *applicable* to a node carrying a set Γ if Γ contains an instance of $\mathcal{N}$. If no rule is applicable, Γ is said to be *saturated*. We call a node σ an *end node* if the set of formulas Γ it carries is saturated, or if $\bot \in \Gamma$. The tableau tree is extended as follows:

1. Choose a leaf node n carrying Γ where n is not an end node, and choose a rule ρ applicable to n.

[2] A similar result was shown in [17] for public announcement enriched with public assignments which are similar to post-conditions.

2. (a) If ρ has one denominator, add the appropriate instanciation to Γ.
   (b) If ρ has k denominators with $k > 1$, create k successor nodes for n, where each successor i carries the union of Γ with an appropriate instanciation of denominator $\mathcal{D}_i$.

A *branch* in a tableau tree is a path from the root to an end node. A branch is *closed* if its end node contains $\bot$, otherwise it is *open*. A tableau tree is *closed* if all its branches are closed, otherwise it is *open*.

For action models without post-conditions, we can use the following two rules: $\dfrac{(\sigma\ \Sigma\ p)}{(\sigma\ \epsilon\ p)}$ $\qquad$ $\dfrac{(\sigma\ \Sigma\ \neg p)}{(\sigma\ \epsilon\ \neg p)}$

These rules are no longer sound in the presence of post-conditions, because the valuation may change after the application of an action model. We propose to replace them by the following rule:

$$\dfrac{(\sigma\ \Sigma\ ;\mathcal{E}, e\ \ \checkmark)}{\begin{array}{c|c} (\sigma\ \Sigma\ \mathsf{post}(e)(p)) & (\sigma\ \Sigma\ \neg\mathsf{post}(e)(p)) \\ (\sigma\ \Sigma\ ;\mathcal{E}\ p) & (\sigma\ \Sigma\ ;\mathcal{E}\ \neg p) \end{array}}\ (post)$$

The complete set of tableau rules is given in Figure 3.

The tableau method contains the classical Boolean rules $(\wedge), (\neg\neg), (\leftarrow_p)$ and $(\leftarrow_{\neg p})$. It also contains the non-deterministic rule $(\neg\wedge)$ handling disjunction. The rule $(\bot)$ makes the current execution fail. The rule for $(\Box_a)$ is applied for all $j \in \{1, \dots i\}$ and all u'_j such that $w'_j R'_a u'_j$. Similarly, the rule for $(\neg\Box_a)$ is applied by choosing non-deterministically for all $j \in \{1, \dots i\}$ some u'_j such that $w'_j R'_a u'_j$ and creating a new fresh label σ_{new}. The rules $(\checkmark)$, $(\otimes)$, $(clash_{\checkmark,\otimes})$ and $(\epsilon_\otimes)$ handle the pre-conditions. Rule $(\checkmark)$ says that if a state survives, then the pre-condition should be true. Rule $(\otimes)$ involves non-determinism: either a state does not survive because the current pre-condition is false or because it did not survive because of a previous pre-condition. Rule $(clash_{\checkmark,\otimes})$ says that a state can not survive and not survive at the same time. Rule $(\epsilon_\otimes)$ says that a state always survives the empty sequence of pointed action models. The last two rules $([\pi \cup \pi'])$ and $(\neg[\pi \cup \pi'])$ handle the union operator.

The rule $(post)$ is non-deterministic and says that either the post condition was true and p is true now or the post condition was false and p is false now.

The proof of soundness and completeness of the tableau is similar to the proof available in [7]. To test whether φ is $\mathcal{AML}^+$-satisfiable, start the tableaux with $\Gamma := \{(\sigma\ \ \epsilon\ \ \varphi)\}$. Given a branch Γ, consider the following tree $\mathcal{T}_\Gamma$:

- Nodes are labels $\sigma \in \mathsf{LAB}$ appearing in Γ;
- Two nodes σ, σ' are linked when a term of the form $(\sigma\ \ R_a\ \ \sigma')$ appears in Γ.

At any step of the algorithm, the depth of $\mathcal{T}_\Gamma$ is linear in the size of φ. The arity of $\mathcal{T}_\Gamma$ is bounded by an exponential in φ since the set of terms of the form $(\sigma\ \ \Sigma\ \ \neg\Box_a\psi)$ is exponential for a given σ. Thus, the set Γ is bounded by an exponential in φ. At any step of the algorithm, we add a new term in Γ. Thus, as Γ requires an exponential amount of space, it takes an exponential amount of

$$\frac{(\sigma\ \Sigma\ \varphi \wedge \psi)}{\begin{array}{c}(\sigma\ \Sigma\ \varphi)\\(\sigma\ \Sigma\ \psi)\end{array}}\ (\wedge) \qquad \frac{(\sigma\ \Sigma\ \neg\neg\varphi)}{(\sigma\ \Sigma\ \varphi)}\ (\neg\neg)$$

$$\frac{(\sigma\ \Sigma\ \neg(\varphi \wedge \psi))}{(\sigma\ \Sigma\ \neg\varphi) \mid (\sigma\ \Sigma\ \neg\psi)}\ (\neg\wedge) \qquad \frac{(\sigma\ \Sigma\ p)(\sigma\ \Sigma\ \neg p)}{\bot}\ (\bot)$$

$$\frac{(\sigma\ \Sigma\ \neg[\mathcal{E}, e]\varphi)}{\begin{array}{l}(\sigma\ \Sigma\ ; \mathcal{E}, e\ \ \checkmark)\\(\sigma\ \Sigma\ ; \mathcal{E}, e\ \ \neg\varphi)\end{array}}\ (\neg[\mathcal{E}, e]) \qquad \frac{(\sigma\ \Sigma\ [\mathcal{E}, e]\varphi)}{(\sigma\ \Sigma\ ; \mathcal{E}, e\ \ \otimes) \ \Bigg|\ \begin{array}{l}(\sigma\ \Sigma\ ; \mathcal{E}, e\ \ \checkmark)\\(\sigma\ \Sigma\ ; \mathcal{E}, e\ \ \neg\varphi)\end{array}}\ ([\mathcal{E}, e])$$

$$\frac{(\sigma\ \Sigma\ ; \mathcal{E}, e\ \ \checkmark)}{\begin{array}{l}(\sigma\ \Sigma\ \mathsf{pre}(e))\\(\sigma\ \Sigma\ \checkmark)\end{array}}\ (\checkmark) \qquad \frac{(\sigma\ \Sigma\ ; \mathcal{E}, e\ \ \otimes)}{\begin{array}{l}(\sigma\ \Sigma\ \checkmark)\\(\sigma\ \Sigma\ \neg\mathsf{pre}(e))\end{array} \ \Bigg|\ (\sigma\ \Sigma\ \otimes)}\ (\otimes)$$

$$\frac{(\sigma\ \Sigma\ ; \mathcal{E}, e\ \ \checkmark)}{\begin{array}{l}(\sigma\ \Sigma\ \mathsf{post}(e)(p))\\(\sigma\ \Sigma\ ; \mathcal{E}\ \ p)\end{array} \ \Bigg|\ \begin{array}{l}(\sigma\ \Sigma\ \neg\mathsf{post}(e)(p))\\(\sigma\ \Sigma\ ; \mathcal{E}\ \ \neg p)\end{array}}\ (post)$$

$$\frac{\begin{array}{l}(\sigma\ \ \mathcal{E}_1, e_1; \dots ; \mathcal{E}_i, e_i\ \ \Box_a\varphi)\\(\sigma\ R_a\ \sigma_1)\end{array}}{\begin{array}{l}(\sigma_1\ \ \mathcal{E}_1, e'_1; \dots ; \mathcal{E}_i, e'_i\ \ \checkmark)\\(\sigma_1\ \ \mathcal{E}_1, e'_1; \dots ; \mathcal{E}_i, e'_i\ \ \varphi)\end{array} \ \Bigg|\ (\sigma_1\ \ \mathcal{E}_1, e'_1; \dots ; \mathcal{E}_i, e'_i\ \ \otimes)}\ (\Box_a)$$

$$\frac{(\sigma\ \ \mathcal{E}_1, e_1; \dots ; \mathcal{E}_i, e_i\ \ \neg\Box_a\varphi)}{\begin{array}{l}(\sigma\ R_a\ \sigma_{\mathrm{new}})\\(\sigma_{\mathrm{new}}\ \ \mathcal{E}_1, e'_1; \dots ; \mathcal{E}_i, e'_i\ \ \checkmark)\\(\sigma_{\mathrm{new}}\ \ \mathcal{E}_1, e'_1; \dots ; \mathcal{E}_i, e'_i\ \ \neg\varphi)\end{array}}\ (\neg\Box_a)$$

$$\frac{(\sigma\ \Sigma\ \otimes)(\sigma\ \Sigma\ \checkmark)}{\bot}\ (clash_{\checkmark,\otimes}) \qquad \frac{(\sigma\ \epsilon\ \otimes)}{\bot}\ (\epsilon_{\otimes})$$

$$\frac{(\sigma\ \Sigma\ [\pi \cup \pi']\varphi)}{\begin{array}{l}(\sigma\ \Sigma\ [\pi]\varphi)\\(\sigma\ \Sigma\ [\pi']\varphi)\end{array}}\ ([\pi \cup \pi']) \qquad \frac{(\sigma\ \Sigma\ \neg[\pi \cup \pi']\varphi)}{\begin{array}{l}(\sigma\ \Sigma\ \neg[\pi]\varphi) \mid\\(\sigma\ \Sigma\ \neg[\pi']\varphi)\end{array}}\ (\neg[\pi \cup \pi'])$$

Figure 3: Tableau rules for $\mathcal{AML}^+$

time to fill Γ until Γ is saturated or until we reach a contradiction. The resulting tableau method can still be turned into a non-deterministic algorithm running in exponential time. $\Box$

Then we can state:

Corollary 6.11. *The satisfiability problem for* $\mathcal{ML}(\mathsf{cp}, \mathsf{rm}^1)$ *is in* NEXPTIME.

As there is a polynomial translation from dynamic epistemic modal logic without post-conditions $\mathcal{AML}$ into $\mathcal{ML}(\mathsf{cp}, \mathsf{rm}^1)$ that preserves satisfiability (see Definition 5.1 and Theorem 5.2), and the satisfiability problem in $\mathcal{AML}$ is NExpTime-hard [7], the satisfiability problem of a formula in $\mathcal{ML}(\mathsf{cp}, \mathsf{rm}^1)$ is NExpTime-hard.

Theorem 6.12. *The satisfiability problem for $\mathcal{ML}(\mathsf{cp}, \mathsf{rm}^1)$ is* NExpTime-*complete.*

7. Related Work

In [10], a dynamic epistemic logic is presented with model changing modalities called 'arrow updates'. Consider an arrow for agent a, i.e., a pair (w, v) in the accessibility relation for agent a, that satisfies a logical condition φ in w (at the start of the arrow) and a condition ψ in v (at the end of the arrow). Then, given a model $\mathcal{M}$, an 'arrow update' for this model *preserves* all pairs $(w, v) \in R_a$ such that $\mathcal{M}, w \models \varphi$ and $\mathcal{M}, v \models \psi$; thus it deletes all other arrows (including for all other agents $b \neq a$). Using our notation, *one* arrow update (such as in [10]) with $(\varphi?, a, \psi?)$, *preserving* all such arrows, clearly corresponds to *three* simultaneous rm^1 actions (length-one path removals) $(\neg\varphi?, a, \neg\psi?)$, $(\neg\varphi?, a, \psi?)$, and $(\varphi?, a, \neg\psi?)$. Now we do not contemplate parallel rm actions, but, as we are only considering Boolean formulas φ and ψ in this submission, the three parallel rm actions correspond to three consecutive rm actions $(\neg\varphi?, a, \neg\psi?)$, $(\neg\varphi?, a, \psi?)$, and $(\varphi?, a, \neg\psi?)$. Dually, each $\mathsf{rm}(()\varphi?, a, \psi?)$ (equally) corresponds to three (simultaneous or consecutive) arrow updates $(\neg\varphi?, a, \neg\psi?)$, $(\neg\varphi?, a, \psi?)$, and $(\varphi?, a, \neg\psi?)$; in [10], the semantic primitive is a finite set of arrows to preserve, simultaneity is here the norm.

Unlike us, [10] does not give complexity results but focus on expressivity and succinctness. Unlike us, [10] do not restrict the updates to those with Boolean pre-conditions but have a fully inductively defined logical language: the φ and ψ tests on the removed arrows can be on any formula. Their reason to focus on preservation instead of removal is, that this is according to the dynamic epistemic logic method that also originally focused on preservation. For example, in public announcement of $!\varphi$ we *preserve* (so not remove) all $(\varphi?, a, \varphi?)$ for all agents a simultaneously. In a further generalization of their work [9] they consider epistemic actions consisting of Kripke frames and where each point in the frame carries a preservation condition (or conditions) $(\varphi?, a, \psi?)$: this then models partial observability. We achieve a similar effect by the interaction of cp and rm. They also demonstrate that arrow updates do not change the expressivity of the language. We can copy that result (however, with the restriction to Boolean tests). However, we have also shown that the expressivity increases once one allows path removal, see Example 4.1. It may further be interesting to observe that our result that an action model with Boolean pre-conditions can be simulated by copy and remove (Theorem 5.2) is a result of the kind called *update equivalence* in [9] (there is an obvious relation to *action emulation* [18]).

In [19] global and local graph modifiers are proposed, where the modifications can concern both the valuations of propositional variables (also known as ontic change, assignment, or substitution) as the accessibility relations. A global graph modification $a - (\varphi, \psi)$ [19, page 295] corresponds to our rm^1 action $(\varphi?, a, \psi?)$, whereas a local graph modification of that kind would amount to a sabotage operator à la van Benthem and as presented in [12, 13, 1, 3, 5], with $\top?$ as the test on beginning and end of the arrow. However, in [19] the only local graph modifiers considered are of the ontic change kind[3]. They consider expressivity of the global version of their logic to other dynamic epistemic logics, in the presence of the universal modality.

Finally, as discussed before, this work should be seen as a continuation of the research developed in [3, 4, 5] on model-changing operations. We recall that one of the model changing operations is the sabotage operator that removes a pair (w, v) from the accessibility relation, without logical conditions in w or v (and sometimes bound by other constraints, such as that w should be the actual world, or that simultaneously to removing (w, v), the actual world shifts from w to v). Now, clearly, 'sabotaging' (w, v) is the same as removing/deleting (w, v). However, there is a very important difference. In the underlying investigation, *all* (w, v) are removed in the rm action $(\varphi?, a, \psi?)$. But in the sabotage action, *a* (arbitrary) (w, v) is removed (without conditions). That makes the latter far more expressive than the former, as this allows one to select (and in that way, so to speak, 'name') an individual arrow, hybrid logic like (as also observed in [19]). Therefore such logics tend to be undecidable and much more expressive, unlike the underlying proposal—we recall that $\mathcal{ML}(\mathsf{cp}, \mathsf{rm}^1)$ is equally expressive than multi-agent modal logic. It is unclear how to bridge the wide gap that jumps straight to undecidability.

8. Conclusion

We proposed the dynamic modal logic $\mathcal{ML}(\mathsf{cp}, \mathsf{rm})$ which contains copy and remove operators. We investigated model theoretic properties of $\mathcal{ML}(\mathsf{cp}, \mathsf{rm})$ such as bisimulations and expressive power. In order to give an appropriate notion of bisimulation, we need the same conditions as for the $\Box^{-1}$ operator, because we need to differentiate states with respect to the paths that traverse them.

We showed that the action model logic $\mathcal{AML}$, one of the best-known dynamic epistemic logics, can be polynomially embedded in $\mathcal{ML}(\mathsf{cp}, \mathsf{rm}^1)$ (the fragment with length-one path removals) when we consider action models with only Boolean pre-conditions. The restriction to Boolean pre-conditions is certainly a limitation, but we discussed examples that show that its expressive power is sufficient to model interesting epistemic scenarios. The embedding

[3] "*In this paper we investigate state label modifiers, leaving edge label modifiers to future work*" [19, page 300].

simulates every finite action model with a combination of copy and remove operators. As we mentioned, the embedding can be done within $\mathcal{ML}(\mathsf{cp}, \mathsf{rm}^1)$ as it only requires single step removals. We showed that the copy and one-step removal themselves correspond to particular action models. As a result we obtain a decomposition method for action models. By decomposing product updates in sequences of copy and remove operators, it would be possible to characterize syntactic fragments of $\mathcal{AML}$ with interesting complexities for the satisfiability problem.

We demonstrated that the complexity of the satisfiability problem of the full language $\mathcal{ML}(\mathsf{cp}, \mathsf{rm})$ is NExpTime-hard. Its upper bound is still open, but we conjecture it is decidable. We proved that satisfiability for the fragment $\mathcal{ML}(\mathsf{rm})$ is decidable, that it is PSpace-complete for $\mathcal{ML}(\mathsf{cp})$ and $\mathcal{ML}(\mathsf{rm}^1)$, and that it is NExpTime-complete for $\mathcal{ML}(\mathsf{cp}, \mathsf{rm}^1)$.

As future work, we plan to extend the analysis of $\mathcal{AML}$ via its embedding in $\mathcal{ML}(\mathsf{cp}, \mathsf{rm})$. In particular, we will address the general case in which action model pre-conditions can be arbitrary formulas of lower complexity. The main challenge when considering the full language is that when pre-conditions are not Boolean, successive applications of the rm operator are no longer independent of each other, and a more involved mapping into $\mathcal{ML}(\mathsf{cp}, \mathsf{rm})$ is required.

Acknowledgments

We acknowledge support from ERC project EPS 313360, from EU 7th Framework Programme under grant agreement n. 295261 (MEALS), and the Laboratoire Internationale Associé “INFINIS”.

References

[1] J. van Benthem, An essay on sabotage and obstruction, in: D. Hutter, W. Stephan (Eds.), Mechanizing Mathematical Reasoning, Vol. 2605 of Lecture Notes in Computer Science, Springer, 2005, pp. 268–276.

[2] H. van Ditmarsch, W. van der Hoek, B. Kooi, Dynamic Epistemic Logic, Kluwer, 2007.

[3] C. Areces, R. Fervari, G. Hoffmann, Moving arrows and four model checking results, in: L. Ong, R. de Queiroz (Eds.), Logic, Language, Information and Computation, Vol. 7456 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2012, pp. 142–153.

[4] C. Areces, R. Fervari, G. Hoffmann, Swap logic, Logic Journal of the IGPL 22 (2) (2014) 309–332.

[5] R. Fervari, Relation-changing modal logics, Ph.D. thesis, Facultad de Matemática Astronomía y Física, Universidad Nacional de Córdoba, Córdoba, Argentina (March 2014).

[6] A. Baltag, L. Moss, S. Solecki, The logic of public announcements, common knowledge and private suspicions, in: I. Gilboa (Ed.), TARK, 1998, pp. 43–56.

[7] G. Aucher, F. Schwarzentruber, On the complexity of dynamic epistemic logic, in: B. Schipper (Ed.), TARK, 2013, pp. 19–28.

[8] H. van Ditmarsch, T. French, Simulation and information, in: J. Broersen, J.-J. Meyer (Eds.), Knowledge Representation for Agents and Multi-Agent Systems, Vol. 5605 of Lectures Notes on Artificial Intelligence, Springer, 2009, pp. 51–65.

[9] B. Kooi, B. Renne, Generalized arrow update logic, in: K. Apt (Ed.), TARK, 2011, pp. 205–211.

[10] B. Kooi, B. Renne, Arrow update logic, Review of Symbolic Logic 4 (4) (2011) 536–559.

[11] C. Areces, H. van Ditmarsch, R. Fervari, F. Schwarzentruber, Logics with copy and remove, in: U. Kohlenbach, P. Barceló, R. de Queiroz (Eds.), WoLLIC 2014, Vol. 8652 of Lecture Notes in Computer Science, Springer, 2014, pp. 51–65.

[12] C. Löding, P. Rohde, Solving the sabotage game is PSPACE-hard, in: Mathematical Foundations of Computer Science 2003, Vol. 2747 of Lecture Notes in Computer Science, Springer, Berlin, 2003, pp. 531–540.

[13] C. Löding, P. Rohde, Model checking and satisfiability for sabotage modal logic, in: P. Pandya, J. Radhakrishnan (Eds.), Proceedings of Foundations of Software Technology and Theoretical Computer Science, 23rd Conference, Vol. 2914 of Lecture Notes in Computer Science, Springer, 2003, pp. 302–313.

[14] J. Plaza, Logics of public communications, Synthese 158 (2) (2007) 165–179.

[15] P. Blackburn, M. de Rijke, Y. Venema, Modal Logic, Vol. 53 of Cambridge Tracts in Theoretical Computer Science, Cambridge University Press, Cambridge, 2001.

[16] R. Goré, Tableau methods for modal and temporal logics, in: M. D'Agostino, D. Gabbay, R. Hähnle, J. Possega (Eds.), Handbook of tableau methods, 1999, pp. 297–396.

[17] H. van Ditmarsch, A. Herzig, T. De Lima, Public announcements, public assignments and the complexity of their logic, Journal of Applied Non-Classical Logics 22 (3) (2012) 249–273.

[18] J. van Eijck, J. Ruan, T. Sadzik, Action emulation, Synthese 185(1) (2012) 131–151.

[19] G. Aucher, P. Balbiani, L. Fariñas Del Cerro, A. Herzig, Global and local graph modifiers, Electronic Notes in Theoretical Computer Science (ENTCS), Special issue Proceedings of M4M 2007 231 (2009) 293–307.